

# Approximations and Optimal Control for State-dependent Limited Processor Sharing Queues

Varun Gupta

Booth School of Business  
University of Chicago

varun.gupta@chicagobooth.edu

Jiheng Zhang

Department of Industrial Engg. and Logistics Management  
The Hong Kong University of Science and Technology

j.zhang@ust.hk

## Abstract

The paper studies approximations and control of a processor sharing (PS) server where the service rate depends on the number of jobs occupying the server. The control of such a system is implemented by imposing a limit on the number of jobs that can share the server concurrently, with the rest of the jobs waiting in a first-in-first-out (FIFO) buffer. A desirable control scheme should strike the right balance between efficiency (operating at a high service rate) and parallelism (preventing small jobs from getting stuck behind large ones).

We employ the framework of *heavy-traffic* diffusion analysis to devise near optimal control heuristics for such a queueing system. However, while the literature on diffusion control of state-dependent queueing systems begins with a sequence of systems and an exogenously defined drift function, we begin with a finite discrete PS server and propose an axiomatic recipe to explicitly construct a sequence of state-dependent PS servers which then yields a drift function. We establish diffusion approximations and use them to obtain insightful and closed-form approximations for the original system under a static concurrency limit control policy.

We extend our study to control policies that dynamically adjust the concurrency limit. We provide two novel numerical algorithms to solve the associated diffusion control problem. Our algorithms can be viewed as “average cost” iteration: The first algorithm uses binary-search on the average cost and can find an  $\epsilon$ -optimal policy in time  $O\left((\log \frac{1}{\epsilon})^2\right)$ ; the second algorithm uses the Newton-Raphson method for root-finding and requires  $O\left(\log \frac{1}{\epsilon} \log \log \frac{1}{\epsilon}\right)$  time.

Numerical experiments demonstrate the accuracy of our approximation for choosing optimal or near-optimal static and dynamic concurrency control heuristics.

## 1 Introduction

Consider an *emergency room* where doctors, nurses, and diagnostic equipment make up a shared resource for admitted patients. It has been empirically observed that the service rate of such service systems is state-dependent (e.g., [5]). Human operators tend to speed up service when there is congestion. As another example, consider a typical *web server or an online transaction processing system*. In such resource sharing systems, as the number of tasks (also called active threads) concurrently sharing the server increases, the server throughput initially increases due to more efficient utilization of resources. However, as the server switches from one task to another, it needs to make room for the new task’s data in its cache memory by evicting an older task’s data (only to fetch it again later). Without a limit on the number of concurrent tasks, this contention for

the limited memory can lead to a phenomenon called *thrashing* which causes the system throughput to drop drastically (e.g., [2, 7, 12, 13, 21, 40]).

The resource sharing system examples we have described above fall into the category of the so-called *State-dependent Limited Processor Sharing* (Sd-LPS) systems. To specify an Sd-LPS system, we begin with a processor sharing (PS) server whose service rate varies as a function of the number of jobs at the server. For example,

$$\mu(1) = 1, \mu(2) = 1.5, \mu(3) = 1.25, \mu(4) = 1, \mu(5) = 0.75, \dots \quad (1)$$

When there are  $n$  jobs at the PS server, each job gets served at a rate of  $\frac{\mu(n)}{n}$  jobs/second. To ensure efficient operation, we impose a limit on the maximum number of jobs that can be served in parallel. We call this the concurrency limit,  $K$ . Arriving jobs that find the server busy with  $K$  jobs wait in a first-come-first-served (FCFS) buffer. A *static concurrency control policy* is one where the concurrency limit is independent of the state. If the concurrency level can vary with the system state (e.g., the queue length of the FCFS buffer), we call it a *dynamic concurrency control policy*.

To understand the tradeoff involved in choosing the optimal concurrency level, suppose there are 3 jobs in the system described above. Even though the server is capable of serving at an aggregate rate of 1.5 jobs/second by limiting the concurrency level to 2, we may choose to increase the concurrency level to 3 and operate below peak capacity. Why might we want to do that? It is well known that if the job size distribution has high variability, then pure PS outperforms FCFS scheduling by allowing small jobs to overtake large ones. Therefore, it may be beneficial to increase the concurrency level beyond the peak efficiency even if some capacity will be lost. Similarly, for job size distributions with low variability, it may be beneficial to operate at  $K = 1$ . Thus Sd-LPS systems are not “work-conserving” queueing systems.

## Contributions

Naturally, our goal is to choose the ‘best’ concurrency control policy. In this work we aim to develop a diffusion approximation framework for Sd-LPS queues, and to utilize the proposed diffusion approximation to find concurrency control policies that minimize the mean sojourn time. This immediately leads to the question: *Given that we want to control the state-dependent PS server (exemplified by (1)), what is a ‘meaningful’ asymptotic scaling and diffusion approximation?*

While there are some works on heavy-traffic asymptotics for queues with state-dependent rates, (i) they begin with a sequence of systems with exogenously given limiting drift functions whereas we begin with a discrete PS server of the kind shown in (1), (ii) they are limited to models where the server can only serve one job at a time whereas multiple jobs are processed in parallel by the PS server; and (iii) they only analyze a Jackson network type of system and do not solve a diffusion control problem. The present paper fills these gaps in the literature.

Our main contributions are as follows:

1. We propose an axiomatic approach to “reverse-engineer” a sequence of Sd-LPS queueing systems starting with a discrete state-dependent PS server (Section 2). This sequence yields a limiting state-dependent drift function which we utilize to develop diffusion approximations. All prior literature on diffusion analysis of state-dependent queues assumes that the drift function is given exogeneously.
2. We propose an approximation for the distribution of the number of jobs in the Sd-LPS

system for a static concurrency limit under a  $GI$  arrival process and  $GI$  job sizes. This approximation is used to choose a near-optimal static concurrency limit to minimize any cost that is a function of the number of jobs in the system.

3. We extend our framework by proposing a more general scaling for developing dynamic (state-dependent) control policies. We present two numerical algorithms for solving the resulting diffusion control problem to minimize the steady-state mean number of jobs in the system. Our algorithms can be viewed as ‘average cost iteration’ (as opposed to value function iteration or policy iteration) and are novel to the best of our knowledge. In our simulation experiments, the dynamic policies based on diffusion control perform remarkably close to the true optimal dynamic policies (for input distributions where the true optimal policy can be computed numerically).

## Related work on control of LPS systems

The literature on LPS-type systems has mostly focused on the constant rate LPS queue where the server speed is independent of the state. Yamazaki and Sakasegawa [42] show qualitatively the effect of increasing the concurrency level on the mean sojourn time for NWU (New-Worse-than-Used) and Erlang job size distributions. Avi-Itzhak and Halfin [4] derive an approximation for the mean sojourn time for the constant rate LPS queue with  $M/GI/$  input process, while Zhang and Zwart [43] derive one for  $GI/GI/$  input. Nair et al. [34] expose the power of LPS scheduling by analyzing the tail of sojourn time under light-tailed and heavy-tailed job size distributions. They prove that with an appropriate choice of the concurrency level as a function of the load, LPS queues can achieve robustness to the distribution of job sizes (their tail to be precise).

For Sd-LPS queues, Rege and Sengupta [38] derive expressions for the moments and distribution of the sojourn time under  $M/M/$  input. Gupta and Harchol-Balter [17] propose an approximation for the mean sojourn time for  $GI/GI/$  input by approximating the interarrival times and job size distribution by the tractable degenerate hyperexponential distribution. They also propose heuristic dynamic admission control policies under  $M/GI/$  input.

In this paper, we propose the first diffusion approximation for Sd-LPS queues with a  $GI/GI$  input and a static concurrency level. In addition, we propose the first heuristic dynamic admission control policies for Sd-LPS queues.

## Related work on control of queueing systems

There is a considerable literature on the control of the arrival and service rates of queueing systems, but the majority of this work focuses on control of  $M/M/1$  or  $M/M/s$  systems via Markov decision process formulation, e.g., [1, 3, 16, 31]. Ward and Kumar [39] look at the diffusion control formulation for admission control in a  $GI/GI/1$  with impatient customers. Our model differs significantly from those in the literature: in our model, the space of actions is the number of jobs admitted to the PS server and is therefore state-dependent. The action space for control problems studied in the literature is usually either the probability of admitting jobs or the server speed, neither of which is state-dependent. The state-dependence of the action space means that the value function may not even be monotonic in the state. We establish this result for our problem and present a simple criterion under which monotonicity holds for general control problems with state-dependent action spaces (see proof of Proposition 3). In addition, the rather arbitrary nature of the service rate curve precludes elegant structural results for the optimal value function which leads us to propose novel and efficient numerical algorithms for solving the resulting diffusion control problem.

## Related work on heavy-traffic analysis of systems with state-dependent rates

Our heavy-traffic scaling is most closely related to the recent work of Lee and Puhalskii [29], who analyze a queueing network of FCFS queues in the critically loaded regime and under non-Markovian arrival and service processes. Yamada [41] also analyzes Markovian state-dependent queueing networks under a similar scaling of state-dependent service and arrival rates. Whereas [29, 41] assume an exogenously given limiting drift function, we propose a method to calculate it from the finite queueing system which is the object of the control problem. Further, the scheduling policy we consider is Processor Sharing. Other works on analysis of heavy-traffic asymptotics of state-dependent Markovian queues include Krichagina [27], Mandelbaum and Pats [32], Janssen et al. [23].

## Outline

In Section 2 we present details of the Sd-LPS model, introduce the notation used in the paper, and describe our approach towards arriving at the asymptotic regime for diffusion analysis. In Section 3, we present our results on diffusion approximation for the Sd-LPS queue under a static concurrency control policy. We defer the proofs of convergence to the appendix. In Section 4 we turn to dynamic concurrency control policies for the Sd-LPS queue. We set up a diffusion control problem for the limiting diffusion-scaled system, and propose two numerical algorithms to solve the diffusion control problem. We make our concluding remarks in Section 5.

## 2 Model and Diffusion Scaling

### 2.1 Stochastic model and Notation

We begin with a description of the Sd-LPS system for which we want to find the optimal control. Let  $X(t)$  denote the total number of jobs in the system at time  $t$ . The control of such a system is implemented by imposing a concurrency limit  $K$ . Only  $Z(t) = X(t) \wedge K$  jobs are in service and server capacity of  $\mu(Z(t))$  is shared equally among the jobs. The remaining  $Q(t) = (X(t) - K)^+$  jobs wait in a FCFS queue. A job, once in service, stays in service until completion. The rate of the server  $\mu(Z(t))$  is understood to be the speed at which it drains the workload. So the *cumulative service amount* a job in service can receive from time  $s$  to  $t$  is

$$S(s, t) = \int_s^t \psi(Z(\tau)) d\tau, \quad (2)$$

where

$$\psi(z) = \begin{cases} \frac{\mu(z)}{z}, & \text{if } z \geq 1, \\ 0, & \text{if } z = 0. \end{cases} \quad (3)$$

Without loss of generality, we assume that there is no intrinsic limit on the number of jobs the server can serve as we can set the service rate to 0 to model such a limit. Note that for the regular state-independent system whose service rate  $\mu(\cdot)$  is a constant, say 1,  $\frac{\mu(z)}{z}$  in the above will simply become  $1/z$ . The state-dependent service rate makes the system *non-work-conserving*, which brings a fundamental challenge to their study. Existing studies of PS or LPS systems crucially rely on the fact that the system is work-conserving, which implies that the workload process is equivalent to that of a simple  $G/G/1$  queue. However, this is not the case for our Sd-LPS model.

The number of job arrivals in time  $[0, t]$  is denoted by  $\Lambda(t)$ . We assume that  $\Lambda(\cdot)$  is a renewal process with rate  $\lambda$ , and  $c_a^2$  denotes the squared coefficient of variation (SCV) for the *i.i.d.* inter-arrival

times. The system is allowed to be non-empty initially. We index jobs by  $i = -X(0) + 1, -X(0) + 2, \dots, 0, 1, \dots$ . The first  $X(0)$  jobs are initially in the system, with jobs  $i = -X(0) + 1, \dots, -Q(0)$  in service and jobs  $i = -Q(0) + 1, \dots, 0$  waiting in the queue. Arriving jobs are indexed by  $i = 1, 2, \dots$ . The size of the  $i$ th job is denoted by  $v_i$ . We assume job sizes are *i.i.d.* random variables with mean size  $m$  (in the chosen unit measuring work) and SCV  $c_s^2$ . Jobs leave the system once the cumulative amount of service they have received from the server exceeds their job sizes.

In this study, we are interested in how the system performance (e.g., expected number of jobs in steady state) depends on the state-dependent service rate function  $\mu(\cdot)$ , the parameters  $(\lambda, c_a^2, m, c_s^2)$  of the stochastic primitives, and the concurrency level  $K$ , which is a decision variable we can control and optimize.

### Measure-valued state descriptor

Analyzing the stochastic processes underlying the Sd-LPS model with generally distributed service times requires tracking of more information about the system state than just the number of jobs. Following the framework in [44, 45], we introduce a *measure-valued* state descriptor to describe the full state of the system. At any time  $t$  and for any Borel set  $A \subset (0, \infty)$ , let  $\mathcal{Q}(t)(A)$  denote the total number of jobs in the buffer whose job size belongs to  $A$  and  $\mathcal{Z}(t)(A)$  denote the total number of jobs in service whose residual job size belongs to set  $A$ . Thus,  $\mathcal{Q}(\cdot)$  and  $\mathcal{Z}(\cdot)$  are measure-valued stochastic processes. Let  $\delta_a$  denote the Dirac measure of point  $a$  on  $\mathbb{R}$  and  $A + y \doteq \{a + y : a \in A\}$ . By introducing the measure-valued processes, we can characterize the evolution of the system via the following *stochastic dynamic equations*:

$$\mathcal{Q}(t)(A) = \sum_{i=B(t)+1}^{\Lambda(t)} \delta_{v_i}(A), \quad (4)$$

$$\mathcal{Z}(t)(A) = \mathcal{Z}(0)(A + S(0, t)) + \sum_{i=B(0)+1}^{B(t)} \delta_{v_i}(A + S(\tau_i, t)), \quad (5)$$

where  $\tau_i$  is the time when the  $i$ th job starts to receive service and

$$B(t) = \Lambda(t) - Q(t), \quad (6)$$

which can be intuitively interpreted as the index of the last job to enter service by time  $t$ . For any Borel measurable function  $f : \mathbb{R}_+ \rightarrow \mathbb{R}$ , the integral of this function with respect to a measure  $\nu$  is denoted by  $\langle f, \nu \rangle$ . Then, both  $Q(t)$  and  $Z(t)$  can be represented using the measure-valued descriptors:

$$Q(t) = \langle 1, \mathcal{Q}(t) \rangle, \quad Z(t) = \langle 1, \mathcal{Z}(t) \rangle.$$

Let  $W(t)$  denote the workload of the system at time  $t$  which is defined as the sum of the sizes of all jobs in queue and the remaining sizes of all jobs in service. Due to the varying service rate of the server, the dynamics of the workload process is represented by

$$W(t) = W(0) + \sum_{i=1}^{\Lambda(t)} v_i - \int_0^t \mu(Z(s)) 1_{\{W(s) > 0\}} ds. \quad (7)$$

Again, we can express the workload  $W(t)$  in terms of the measure-valued descriptors:

$$W(t) = \langle \chi, \mathcal{Q}(t) + \mathcal{Z}(t) \rangle, \quad (8)$$

where  $\chi$  denotes the identity function on  $\mathbb{R}$ .

## 2.2 Proposed Asymptotic Regime for Diffusion Approximation of Sd-LPS systems

We refer to the system introduced in Section 2.1 as our *original* system. We now propose an asymptotic regime where a sequence of Sd-LPS systems, parametrized by  $r \in \mathbb{Z}^+$ , will be studied under an appropriate scaling. The objective is to obtain a meaningful approximation of the original system with the goal of choosing the ‘best’ concurrency control policy. This leads to the question:

*What is the appropriate scaling to analyze the Sd-LPS queue? That is, what asymptotic regime captures the entire service-rate curve of the original Sd-LPS system, and thus can be used to find a near-optimal concurrency limit?*

As we mentioned earlier, the scaling we develop is very close to the scaling proposed by Yamada [41] and Lee et al. [29]. To provide an axiomatic justification for why this is the appropriate scaling for Sd-LPS systems we begin by examining *two special cases of Sd-LPS systems*: (i) multiserver systems, and (ii) the constant rate LPS queue.

**The  $G/GI/k$  multiserver system** A  $G/GI/k$  multiserver system with a service rate of  $\mu$  jobs/second per server and a central buffer can be viewed as an Sd-LPS system with  $\mu(n) = n\mu$  and a concurrency limit of  $K = k$ . There is a rich literature on the question of whether having many slow servers is better than having a few fast servers (e.g., Brumelle [9], Daley and Rolski [11]), which is similar in spirit to the concurrency control problem. The work on diffusion approximations for multiserver systems started with Köllerström [25] for the classical heavy-traffic regime where  $k$  and  $\mu$  are held constant while  $\lambda \uparrow k\mu$ . A more refined heavy-traffic regime is the Halfin-Whitt regime (starting with [18] and more recently [37], [14]) where one fixes  $\mu$  and creates a sequence of multiserver systems parametrized by  $r$ , where the number of servers grows according to  $k^{(r)} = rk$  while the mean arrival rate  $\lambda^{(r)}$  increases so that  $\frac{k^{(r)}\mu - \lambda^{(r)}}{\sqrt{k^{(r)}}} \rightarrow \beta$ . The constant  $\beta$  is chosen so that the probability that an arrival gets blocked converges to a non-degenerate limit (bounded away from 0 and 1). An extremely accurate diffusion approximation for a given  $G/M/k$  system can be obtained by matching the blocking probability under the Halfin-Whitt regime.

**State-independent (constant rate) LPS queue** In the state-independent LPS queue, the service rate of the server is a constant  $\mu$  irrespective of the number of jobs at the server, and there is a fixed concurrency limit  $k$ . Recently, Zhang et al. [45] have proposed and analyzed a diffusion approximation for the LPS system where a sequence of LPS systems (parametrized by  $r$ ) is devised so that the service rate remains fixed at  $\mu$ , the concurrency limit increases according to  $k^{(r)} = rk$  and the arrival rate increases so that  $k^{(r)}(\mu - \lambda^{(r)}) \rightarrow \theta$ , a constant. As in the Halfin-Whitt regime for the multiserver systems, under the proposed scaling for LPS systems the probability that an arrival finds all slots at the PS server occupied converges to a constant. In addition, the queue length scaled by  $\frac{1}{k^{(r)}}$  also converges to a non-degenerate distribution, unlike Halfin-Whitt where the queue lengths are smaller and must be scaled by  $\frac{1}{\sqrt{k^{(r)}}}$ .

It is not obvious how either of these scalings can be extended to the Sd-LPS system, but among the chief desiderata is that the diffusion-scaled system should in some sense be a *faithful proxy* for the original system. As an example of a regime that is not quite faithful enough, consider the following



example: We scale the concurrency limit as  $k^{(r)} = rk$ , leave the mean arrival rate  $\lambda$  unchanged, and ‘stretch’ the service rate curve so that for the  $r$ th Sd-LPS system,  $\mu^{(r)}(rx) = \hat{\mu}(x)$  where  $\hat{\mu}(\cdot)$  is a continuous interpolation of  $\mu(\cdot)$ . This would correspond to a fluid limit where the steady state ‘gets stuck’ around  $x^*$ , where  $\hat{\mu}(x^*) = \lambda$ . Therefore this fluid regime cannot be used to devise a control policy for the original Sd-LPS system.

Instead, we adopt an axiomatic approach to devising the asymptotic regime: Under some *reasonable non-trivial assumptions*  $\mathcal{A}$ , the *behavior*  $\mathcal{B}$  of the diffusion-scaled system *should mimic* the original discrete system we want to approximate. The choice of  $\mathcal{A}$  and  $\mathcal{B}$  can be seen as the axioms of our scaling which we will use to reverse-engineer a diffusion scaling. We now formalize our choice of the assumptions  $\mathcal{A}$  and behavior  $\mathcal{B}$  for Sd-LPS systems with static concurrency control. Later we will use the intuition gained from this exercise to engineer a scaling for developing dynamic control policies.

### Axioms for the Sd-LPS diffusion scaling

We construct a sequence of Sd-LPS systems parametrized by  $r \in \mathbb{Z}^+$  such that the  $r$ th system has a concurrency level of

$$k^{(r)} = rK. \quad (9)$$

Further, the sequence of service rate curves  $\mu^{(r)}(\cdot)$  satisfies:

( $\mathcal{A}$ , **the assumptions**) under a Poisson arrival process with rate  $\lambda$  and *i.i.d.* Exponentially distributed job sizes with mean size  $m$  (i.e., under  $M/M/$  input),

( $\mathcal{B}$ , **the behavior**) the distribution of the scaled number of jobs in the system (scaled by  $\frac{1}{k^{(r)}}$ ) converges to a non-degenerate limit.

### Consequences of the scaling axioms and an alternate characterization

Since we start by fixing the concurrency levels for the sequence of Sd-LPS systems, the only design flexibility we have to satisfy the scaling axioms is the choice of state-dependent service rate curves. Let us denote the service rate curve for the  $r$ th system by  $\mu^{(r)}(i)$ , and the resulting distribution of the number of jobs in the  $r$ th system under  $M/M/$  input by  $F^{(r)}$ . Our goal is to find the sequence  $\mu^{(r)}(\cdot)$  so that

$$\lim_{r \rightarrow \infty} F^{(r)}(\lceil rx \rceil) = \hat{F}(x) \quad \forall x \in [0, \infty) \quad (10)$$

for some distribution function  $\hat{F}(\cdot)$ . This gives us our **first way** of deriving the scaling: Fix  $\hat{F}(\cdot)$  to be a smooth, strictly increasing interpolation of the distribution function of the original system under  $M/M/$  input and reverse-engineer the sequence of service rate functions  $\mu^{(r)}(i)$ . As we will show in the next section, the requisite service rate functions satisfy

$$\lim_{r \rightarrow \infty} r \left( \lambda m - \mu^{(r)}(\lceil rx \rceil) \right) = \lambda m \frac{d \log f(x)}{dx} \quad \forall x \in [0, \infty). \quad (11)$$

where  $f(x) = \frac{d}{dx} \hat{F}(x)$ .

Of course, by reverse-engineering the scaling, we guarantee ourselves a non-degenerate limit that is interesting in that it captures the effect of the entire  $\mu(\cdot)$  function. Further, it turns out we never really need to compute the service rate functions  $\mu^{(r)}(\cdot)$ ! In Section 3, we will show that we can directly express the limiting steady-state quantities in terms of the distribution  $\hat{F}(\cdot)$ , which can be easily obtained from that of the original system.

However, the method described above does not generalize to dynamic control policies since the distribution  $F(\cdot)$  (and its smooth interpolation  $\hat{F}(\cdot)$ ) was obtained under the assumption of a static concurrency limit of  $K$ . For this case, we propose a **second way** of deriving the scaling, that still guarantees (10):

Begin with  $\hat{\mu}(\cdot) : \mathbb{R}_0^+ \rightarrow \mathbb{R}_0^+$  satisfying:

1.  $\hat{\mu}(\cdot)$  agrees with  $\mu(\cdot)$  at integer arguments:  $\hat{\mu}(i) = \mu(i)$  for  $i = \{1, 2, \dots\}$
2.  $\hat{\mu}(\cdot)$  is continuous and smooth

The sequence of service rate functions  $\{\mu^{(r)}(\cdot)\}$  is chosen to satisfy

$$\lim_{r \rightarrow \infty} r \left( \lambda m - \mu^{(r)}(\lceil rx \rceil) \right) = \lambda m \log \frac{\lambda m}{\hat{\mu}(x)} \quad \forall x \in [0, \infty). \quad (12)$$

For either way of arriving at the diffusion scaling, we see that  $r(\lambda - \mu^{(r)}(\lceil rx \rceil))$  converges to a non-degenerate *drift function*  $-\theta(x)$ . In the first case the  $\theta(x)$  function is reverse-engineered by fixing a limiting distribution, and in the second case it is obtained more directly using a continuous extension of  $\mu(i)$ . The first is more appropriate for finding static control policies, while the second is more appropriate for computing dynamic control policies. In both cases there is limited flexibility in extending a discrete function to a continuous smooth function.

### Intuitive explanation for the choice of service rate curves $\mu^{(r)}(\cdot)$

We begin by explaining our first choice of the asymptotic regime (11). Consider the  $r$ th Sd-LPS system operating under  $M/M/$  input. Let  $\pi^{(r)}(i)$  be the probability mass function for the steady-state number of jobs in the  $r$ th system. Flow-balance equations imply

$$\frac{\pi^{(r)}(\lceil rx + 1 \rceil)}{\pi^{(r)}(\lceil rx \rceil)} = \frac{\lambda m}{\mu^{(r)}(\lceil rx \rceil)}.$$

Since, by design, we want  $r\pi^{(r)}(\lceil rx \rceil)$  to converge to the density function  $f(x)$ , we should have:

$$\frac{\lambda m}{\mu^{(r)}(\lceil rx \rceil)} \approx \frac{f\left(x + \frac{1}{r}\right)}{f(x)} \approx 1 + \frac{1}{r} \frac{f'(x)}{f(x)}.$$

Equivalently,  $r(\lambda m - \mu^{(r)}(\lceil rx \rceil)) \rightarrow \lambda m \frac{d \log f(x)}{dx}$ .

To motivate the second proposal of  $\mu^{(r)}(\cdot)$ , note that if  $\frac{\lambda m}{\mu^{(r)}(\lceil rx \rceil)} \approx 1 - \frac{\theta(x)}{\lambda m r} \approx e^{-\frac{\theta(x)}{\lambda m r}}$ , then

$$\frac{\pi^{(r)}(ry)}{\pi^{(r)}(rx)} \approx e^{-\frac{1}{\lambda m} \int_x^y \theta(u) du} \rightarrow \frac{\pi(y)}{\pi(x)} = \prod_{i=x+1}^y \frac{\lambda m}{\mu(i)}.$$

Or,

$$-\frac{1}{\lambda m} \int_x^y \theta(u) du \approx \log \frac{\pi^{(r)}(ry)}{\pi^{(r)}(rx)} \approx \log \frac{\pi(y)}{\pi(x)} = \sum_{i=x+1}^y \log \frac{\lambda m}{\mu(i)}.$$

Comparing the first and last expressions above gives us an approximation for  $\theta(u)$  in terms of a continuous extension  $\hat{\mu}$  of  $\mu$ :  $r(\lambda m - \mu^{(r)}(\lceil rx \rceil)) \rightarrow -\lambda m \log \frac{\lambda m}{\hat{\mu}(x)}$ .



**Comparison with existing diffusion scalings** For the two specific examples of Sd-LPS systems we pointed out earlier we can now ask: What axioms do the existing diffusion scalings satisfy?

The **implicit** axioms used by [18] posit that each system in the sequence is itself a homogeneous multiserver system, and further, under  $M/M/$  input the blocking probability of the sequence converges to a non-degenerate limit (for example, to the blocking probability of the finite system being approximated). If we use our proposed scaling to approximate a multiserver system, the sequence of Sd-LPS systems would not be a homogeneous multiserver system. Indeed,  $r(\lambda m - \mu^{(r)}(rx))$  grows as  $\log(x)$  for small  $x$ . On the positive side, this flexibility allows us to capture the entire distribution of number of jobs, not just the blocking probability. We should also point out that while the Halfin-Whitt scaling is more useful for capacity provisioning, the goal of our proposed scaling is to solve the admission/concurrency control problem.

For the constant rate LPS system, our scaling matches the diffusion scaling of Zhang et al. [45], and thus can be seen as an extension of their scaling to Sd-LPS. Our convergence proofs follow their outline as well.

### 3 Diffusion approximation for the Sd-LPS queue with a static concurrency level

The goal of this section is to provide approximations for the steady-state performance of the Sd-LPS queue with a static concurrency level under the proposed scaling (11). In Section 3.1 we first summarize the results of this section by giving an approximation for the mean number of jobs in an Sd-LPS system under a static concurrency level (equation (13)), and providing some simulation results which show the utility of the approximation for choosing a near-optimal concurrency level. In Section 3.2, we prove process-level limits for diffusion-scaled workload and head count processes. In Section 3.3, we justify using the steady state of the limiting processes as an approximation for the limit of the steady state of the diffusion-scaled processes by establishing the required interchange of limits. We also present closed-form formulae for these steady-state distributions. All the proofs for this section can be found in the appendix.

#### 3.1 An approximation and simulation results

Let  $N$  denote the steady-state number of jobs in the Sd-LPS system for a given static concurrency level  $K$ . Our main result of this section yields the following simple approximation formula for the expectation of  $N$  as a function of the concurrency level and other system parameters (see Proposition 2 for the formal statement)

$$\mathbf{E}[N] \approx \frac{\sum_{n=0}^{\infty} (n \wedge K) \pi(n)^{\frac{c_s^2+1}{c_s^2+c_a^2}}}{\sum_{n=0}^{\infty} \pi(n)^{\frac{c_s^2+1}{c_s^2+c_a^2}}} + \left( \frac{c_s^2+1}{2} \right) \frac{\sum_{n=0}^{\infty} (n-K)^+ \pi(n)^{\frac{c_s^2+1}{c_s^2+c_a^2}}}{\sum_{n=0}^{\infty} \pi(n)^{\frac{c_s^2+1}{c_s^2+c_a^2}}}, \quad (13)$$

where  $\pi(n)$  denotes the steady-state probability of there being  $n$  jobs in the Sd-LPS system under  $M/M/$  input (that is, Poisson arrivals with mean rate  $\lambda$  and *i.i.d.* Exponentially distributed job sizes with mean size  $m$ ).

Figure 1 shows a hypothetical service rate function for a PS server. The service rate has the functional form  $\mu(i) = 1.25 - \frac{i^2}{150}$ , and is monotonically decreasing in the concurrency level. Figure 2 shows the simulation results for the steady-state mean number of jobs as a function of the concurrency level  $K$ . The arrival process is Poisson with mean arrival rate shown below the figures.

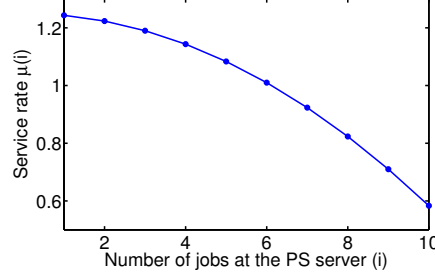


Figure 1: State-dependent service rate function used for simulation results

We simulated three distributions, each with mean  $m = 1$  and SCV  $c_s^2 = 19$ . The solid curve shows the diffusion approximation (13) for the mean number of jobs. For each value of  $\lambda$  and each distribution, the optimal concurrency level obtained via approximation (13) matches the one obtained from simulating the LPS system. We should point out the caveat that while the proposed diffusion approximation accurately captures the *shape* of  $\mathbf{E}[N]$  versus the concurrency level curve and thus provides good guidance for concurrency control, the actual numerical values for  $\mathbf{E}[N]$  are not always very accurate for all values of  $K$ .

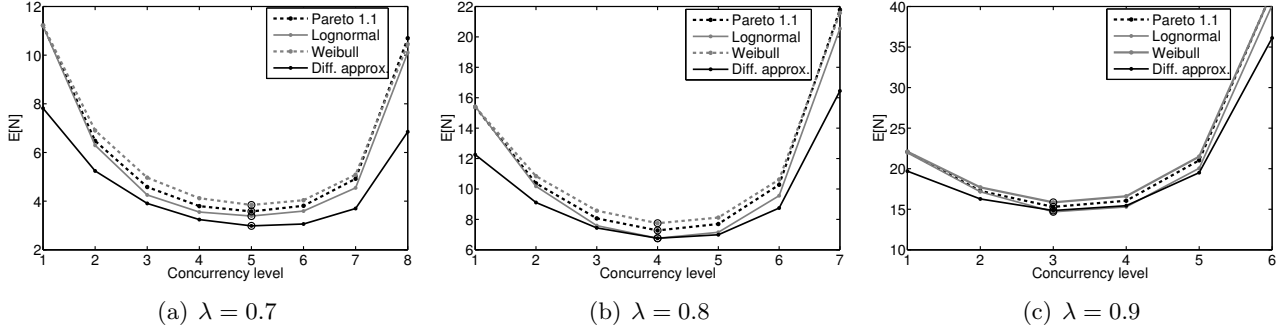


Figure 2: Simulation results for mean number of jobs in the system versus the concurrency level for the service rate function shown in Figure 1 for various job size distributions, all with mean  $m = 1$  and SCV  $c_s^2 = 19$ . The arrival process is Poisson with indicated mean arrival rate  $\lambda$ . Also shown is the diffusion approximation from equation (13). The optimal concurrency level for each curve is shown with a circle.

### 3.2 Diffusion analysis of Sd-LPS system

We now present the analysis of the Sd-LPS system under the asymptotic regime described in (11). For generality and notational convenience, we present all the analysis in terms of the general drift function  $\theta(x)$ , and then translate the result into a form involving  $f(x)$  (Proposition 2) for convenience.

Consider the sequence of Sd-LPS systems indexed by  $r$ . We append a superscript  $(r)$  to all the quantities associated with the  $r$ th system. The concurrency level  $k^{(r)}$  is specified as (9).

Assume that the arrival process  $\Lambda^{(r)}(\cdot)$  satisfies

$$\frac{\Lambda^{(r)}(r^2t) - r^2\lambda t}{r} \Rightarrow M_a(t), \quad \text{as } r \rightarrow \infty, \quad (14)$$

where  $M_a(\cdot)$  is a Brownian motion with zero drift and variance  $c_a^2$ . Further, we assume that the sizes of arriving jobs follow distribution  $G$  which satisfies

$$G \text{ is a continuous distribution function with mean } m. \quad (15)$$

Introduce the drift function

$$\theta^{(r)}(x) = \begin{cases} r(\mu^{(r)}(\lceil rx \rceil) - \lambda m) & x > 0, \\ 0 & x = 0. \end{cases}$$

The above definition is only for technical convenience, since otherwise  $\mu^{(r)}(0)$  would be undefined. However, this does not matter since the server idles when there are no jobs in the system. The heavy traffic condition is specified by

$$\theta^{(r)}(x) \xrightarrow{u.o.c} \theta(x) \quad \text{as } n \rightarrow \infty, \quad (16)$$

for some locally Lipschitz continuous function  $\theta(\cdot)$  on  $(0, \infty)$  satisfying

$$\theta(K) > 0. \quad (17)$$

The notation  $\xrightarrow{u.o.c}$  means uniform convergence on compact sets, which is only required for technical reasons. Condition (17) ensures that the system is stable (see the proof of Theorem 2). As a quick remark, we make a connection with the traditional single server system where the server speed is constant, say  $\mu^{(r)}(\cdot) \equiv 1$ , and the drift is created by constructing a sequence of  $\lambda^{(r)}$  which converges to  $\lambda$  at the rate of  $1/r$ . The heavy traffic condition for this constant rate LPS system then becomes

$$r(1 - \lambda^{(r)}m) \rightarrow \theta > 0, \quad \text{as } r \rightarrow \infty.$$

We are interested in the asymptotic behavior of the diffusion-scaled processes for the  $r$ th system, defined as

$$\hat{X}^{(r)}(t) = \frac{1}{r}X^{(r)}(r^2t), \quad \hat{W}^{(r)}(t) = \frac{1}{r}W^{(r)}(r^2t). \quad (18)$$

The diffusion scaling for other stochastic processes  $\mathcal{Q}^{(r)}$ ,  $\mathcal{Z}^{(r)}$ ,  $Z^{(r)}$ ,  $Q^{(r)}$  and  $B^{(r)}$  is defined in the same way. To obtain the diffusion limit of the head count process  $\hat{X}^{(r)}$  and workload process  $\hat{W}^{(r)}$ , we need to carefully analyze the measure-valued processes introduced. The detailed analysis is presented in Appendix B.

Since we need to work with the measure-valued process, let  $\nu$  denote the probability measure associated with the probability distribution function  $G$ , and  $\nu_e$  denote the probability measure associated with the *equilibrium* distribution  $G_e$  of  $G$ . That is,  $G_e(x) = \frac{1}{m} \int_0^x [1 - G(y)] dy$  and the mean of  $G_e$  is

$$m_e = \frac{1 + c_s^2}{2} m.$$

Let  $\mathbf{M}$  denote the space of all non-negative finite Borel measures on  $[0, \infty)$ . We need the following regularity assumptions on the initial state to rigorously prove the diffusion approximation results. Assume there exists  $(\xi^*, \mu^*) \in \mathbf{M} \times \mathbf{M}$  such that

$$(\hat{Q}^{(r)}(0), \hat{Z}^{(r)}(0)) \Rightarrow (\xi^*, \mu^*), \quad (19)$$

$$\langle \chi^{1+p}, \hat{Q}^{(r)}(0) + \hat{Z}^{(r)}(0) \rangle \Rightarrow \langle \chi^{1+p}, \xi^* + \mu^* \rangle \quad \text{for some } p > 0, \quad (20)$$

as  $r \rightarrow \infty$ , and

$$(\xi^*, \mu^*) = \left( \frac{w^* \wedge K m_e}{m_e} \nu, \frac{(w^* - K m_e)^+}{m} \nu_e \right), \quad (21)$$

where  $w^* = \langle \chi, \xi^* + \mu^* \rangle$ . The above regularity assumptions (19)–(21) basically require that the sequence of initial states is well behaved. These assumptions, together with the heavy traffic assumptions (14)–(16), are made throughout the rest of this paper.

The first result we present is an asymptotic relationship, called State Space Collapse (SSC), between the workload process and the head count process. Define a map  $\Delta_K(\cdot) : \mathbb{R}_+ \rightarrow \mathbb{R}_+$  by

$$\Delta_K(w) = \frac{w \wedge K m_e}{m_e} + \frac{(w - K m_e)^+}{m}. \quad (22)$$

The SSC result states that the total number of jobs in the system  $\hat{X}^{(r)}$  can be *asymptotically* represented using the workload  $\hat{W}^{(r)}$  via the map  $\Delta_K$ , which is a bijective map meaning that workload can also be represented using the total number of jobs. SSC is described as follows:

**Proposition 1 (State Space Collapse)** *For the sequence of Sd-LPS systems parametrized by  $r \in \mathbb{Z}^+$  and satisfying initial conditions (19)–(21), as  $r \rightarrow \infty$ ,*

$$\sup_{t \in [0, T]} |(\hat{X}^{(r)}(t) \wedge K) m_e + (\hat{X}^{(r)}(t) - K)^+ m - \hat{W}^{(r)}(t)| \Rightarrow 0. \quad (23)$$

Note that

$$\Delta_K^{-1}(x) = (x \wedge K) m_e + (x - K)^+ m$$

is the inverse of the map  $\Delta_K(\cdot)$ . A full version of the SSC, which demonstrates a bijective map between the workload  $\hat{W}^{(r)}$  and the measure-valued status  $(\hat{Q}^{(r)}, \hat{Z}^{(r)})$ , is presented and proved in Appendix B. Roughly speaking, SSC reveals that the residual sizes of jobs in service follow the equilibrium distribution  $G_e$ . The simpler SSC of Proposition 1 can be derived from the full version proved in Appendix B. For the purpose of performance analysis and for optimal control in this paper, we only need the simple version of SSC.

The next step is the analysis of the workload process defined in (7). The challenge here is that the evolution of workload depends on the number of jobs in service due to the state-dependent service rate. The simple SSC result allows us to overcome this difficulty. The following theorem establishes the diffusion limit of the workload process  $\hat{W}^{(r)}(t)$  and the number of jobs  $\hat{X}^{(r)}$  as reflected Brownian motion (RBM) with state-dependent drifts.

**Theorem 1 (Weak convergence to RBMs with state-dependent drift)** *For the sequence of Sd-LPS systems parametrized by  $r \in \mathbb{Z}^+$  satisfying (19)–(21), as  $r \rightarrow \infty$ ,*

$$\hat{W}^{(r)} \Rightarrow W^*, \quad (24)$$

where  $W^*$  is an RBM with initial value  $W^*(0) = w^*$ , drift  $-\theta(\Delta_K(W^*(t)) \wedge K)$  and variance  $\sigma^2 = \lambda m^2(c_a^2 + c_s^2)$ . Moreover, as  $r \rightarrow \infty$ ,

$$\hat{X}^{(r)} \Rightarrow X^* = \Delta_K(W^*). \quad (25)$$

The proof of Theorem 1 is presented in Appendix B.

### 3.3 Steady State of the Diffusion Limit

The entire goal of heavy traffic analysis is to obtain a tractable process, an RBM with state-dependent drift, as an approximation of the complicated stochastic process underlying the original model. That is, the steady state of the limiting RBM can be computed. We begin with a basic result on the steady-state distribution of an RBM with state-dependent drift and variance (Lemma 1). We then use this lemma to derive the steady-state distribution and mean of the limiting workload and number of jobs ( $W^*$  and  $X^*$ ) in terms of the primitives of the original Sd-LPS model.

**Lemma 1** *The stationary distribution of a one-dimensional RBM,  $W$ , with state-dependent drift  $-\beta(\cdot)$  and state-dependent variance  $s(\cdot)$  is given by*

$$\Pr[W(\infty) \leq w] = \alpha \int_0^w e^{-\int_0^u \frac{\beta(v) + \frac{1}{2}s'(v)}{\frac{1}{2}s(v)} dv} du = \alpha \int_0^w \frac{1}{s(u)} e^{-\int_0^u \frac{\beta(v)}{\frac{1}{2}s(v)} dv} du, \quad (26)$$

where  $\alpha$  is a normalization constant.

The proof of this lemma is presented in Appendix B. In our setting, the drift  $\beta(w) = \theta(\Delta_K(w))$  and the variance  $s(w) = \sigma^2 = \lambda m^2(c_a^2 + c_s^2)$  is a constant. Using Lemma 1, we immediately have the closed-form expression for the steady-state  $W^*(\infty)$  of the diffusion limit  $W^*$  in terms of the drift function  $\theta(\cdot)$ :

**Corollary 1** *With  $W^*$  as defined in Theorem 1,*

$$\Pr[W^*(\infty) \leq w] = \begin{cases} \alpha \int_0^w e^{-\frac{2 \int_0^u \theta(v/m_e)}{\sigma^2} dv} du & w \leq K \cdot m_e, \\ \alpha \int_0^w \left( e^{-\frac{2 \int_0^{K m_e} \theta(v/m_e)}{\sigma^2} dv} \right) e^{-\frac{2\theta(K)(u - K m_e)}{\sigma^2}} du & w > K \cdot m_e. \end{cases} \quad (27)$$

To present the result for the limiting steady-state quantities in a form that is easier to apply in practice, we express the drift function  $\theta(\cdot)$  as follows:

$$-\theta(x) = \lambda m \frac{d \log f(x)}{dx} \quad (28)$$

where, recall, the function  $f(\cdot)$  represents the derivative of a twice-differentiable interpolation of the distribution of steady-state number of jobs for the original Sd-LPS system with concurrency level  $K$  under  $M/M/$  input (see the discussion on the asymptotic regime in Section 2.2 preceding equation (11)). We assume that  $\frac{d \log f(x)}{dx}$  is a constant less than 0 on the interval  $[K, \infty)$  (it is easy to verify that such an extension exists). It turns out that while we need  $f(\cdot)$  to be differentiable to define  $\theta(\cdot)$ , the limiting steady-state distribution is well defined even without this condition. Finally, we obtain the following result on the steady-state distribution of workload and number of jobs in the system:

**Proposition 2** *Let  $W^*$  and  $X^*$  be the workload and number of jobs for the limiting Sd-LPS system (as defined in Theorem 1). Let the drift function  $\theta(x)$  be given by  $-\theta(x) = \lambda m \frac{d \log f(x)}{dx}$ .*

*The steady-state distributions of  $W^*$  and  $X^*$  are given by*

$$\Pr[W^*(\infty) \leq w] = \alpha \int_0^{\frac{w}{me}} f(x) \frac{c_s^2+1}{c_s^2+c_a^2} dx, \quad (29)$$

$$\Pr[X^*(\infty) \leq x] = \begin{cases} \alpha \int_0^x f(u) \frac{c_s^2+1}{c_s^2+c_a^2} du & x \leq K, \\ \alpha \int_0^{K+(x-K)\frac{m}{me}} f(u) \frac{c_s^2+1}{c_s^2+c_a^2} du & x > K, \end{cases} \quad (30)$$

where  $\alpha$  is the normalization constant. The mean of the limiting scaled number of jobs is given by

$$\mathbf{E}[X^*(\infty)] = \frac{\int_{x=0}^{\infty} (x \wedge K) f(x) \frac{c_s^2+1}{c_s^2+c_a^2} dx}{\int_{x=0}^{\infty} f(x) \frac{c_s^2+1}{c_s^2+c_a^2} dx} + \frac{c_s^2+1}{2} \cdot \frac{\int_{x=0}^{\infty} (x-K)^+ f(x) \frac{c_s^2+1}{c_s^2+c_a^2} dx}{\int_{x=0}^{\infty} f(x) \frac{c_s^2+1}{c_s^2+c_a^2} dx}. \quad (31)$$

We have thus obtained closed-form formulae (approximations) for steady-state quantities based on the limiting diffusion process. The approximation (13) at the beginning of this section is obtained from (31) by further using the probability mass function,  $\pi(\cdot)$ , for the number of jobs corresponding to the original Sd-LPS system in place of the density function  $f(\cdot)$ .

We now close the loop by translating the convergence at the process level to convergence of steady-state distributions in the following theorem. The proof is presented in Section 2. This justifies the formulae in Proposition 2 as an approximation for the steady state of the original Sd-LPS system. The quality of the approximation is demonstrated in the numerical experiment presented at the beginning of this section (see Figure 2).

**Theorem 2 (Convergence of steady-state distributions)** *For all large enough  $r$ , the stochastic process  $\hat{X}^{(r)}$  has a steady state, denoted by  $\hat{X}^{(r)}(\infty)$ . Moreover,*

$$\begin{aligned} \hat{W}^{(r)}(\infty) &\Rightarrow W^*(\infty), \\ \hat{X}^{(r)}(\infty) &\Rightarrow X^*(\infty), \end{aligned}$$

where  $W^*(\infty)$  and  $X^*(\infty)$  are characterized in (27) and (30).

## 4 Dynamic concurrency control for the Sd-LPS queue

In Section 3, we established approximations for the steady-state number of jobs and workload in an Sd-LPS system operating under a *static* concurrency level. Our numerical experiments showed that the optimal static level based on the approximations yields near-optimal performance for the original Sd-LPS system. In this section we go further by allowing a *dynamically adjustable* concurrency level.

In Section 4.1 we first summarize our translation of the discrete state space control problem for the original system to a continuous state space diffusion control problem, and the translation of the resulting control back to that for the original Sd-LPS system. To demonstrate the efficacy of our approach, we present results of numerical experiments comparing the performance of the proposed diffusion limit based control policies against the true optimal dynamic control policy for a special non-trivial input process for which the true optimal policy can be computed numerically.

In Section 4.2 we formulate the diffusion control problem and show how this can help solve the dynamic control problem for the original system. We then describe two novel numerical algorithms to solve the diffusion control problem: an algorithm that iteratively refines its estimate of the average cost of the optimal policy using binary search in Section 4.3, and an algorithm that uses the Newton-Raphson root finding method to search for the average cost of the optimal policy in Section 4.4.

#### 4.1 Overview of our approach and Simulation results

The following steps outline our approach to obtaining a heuristic dynamic control policy for the original Sd-LPS server:

1. Convert the discrete service rate vector  $\mu(i)$  for the original state-dependent PS server into a drift function according to (12):

$$\theta(x) \doteq -\lambda m \log \frac{\lambda m}{\hat{\mu}(x)},$$

where  $\hat{\mu}$  is a continuous extension of  $\mu(i)$ .

2. Formulate a diffusion control problem to minimize the steady-state mean number of jobs. The action/control will be the concurrency level as a function of the state. For convenience, we frame the diffusion control problem with workload as the state variable since the variance of workload is a constant (i.e., independent of state or action), and the control affects the drift of the workload through  $\theta(x)$ .
3. Given  $k^*(w)$ , the optimal concurrency level as a function of the workload for the diffusion control problem, to obtain a control policy for the original (discrete) Sd-LPS system, we first obtain a control function  $\tilde{k}(w)$  with discrete concurrency levels by rounding  $k(w)$  to the nearest integer for all  $w$ . The control algorithm is implemented by using  $\tilde{W}(t) = m_e Z(t) + mQ(t)$  for the original system as the proxy for the current workload, and then taking action to reach the concurrency level dictated by the diffusion control problem:  $\tilde{k}(\tilde{W}(t))$ . In controlling the original system we only take actions upon job arrivals and departures, do not preempt jobs once they enter service, and do not increase the concurrency level by more than one in any arrival/departure event. The precise policy is given as follows:

- **On arrival at  $t$ :** Let  $\tilde{W} = m_e Z(t_-) + m(Q(t_-) + 1)$ , where  $(Z(t_-), Q(t_-))$  denotes the system state immediately before the event. If  $\tilde{k}(\tilde{W}) \geq (Z(t_-) + 1)$  then admit one job to the server at  $t$ , otherwise do nothing.
- **On departure at  $t$ :** Let  $\tilde{W} = m_e (Z(t_-) - 1) + mQ(t_-)$ . Admit  $\min \{(\tilde{k}(\tilde{W}) - Z(t_-) + 1)^+, 2\}$  jobs at  $t$ .

#### Simulation Results

Table 1 shows experimental results comparing the performance of the dynamic policies produced using the proposed diffusion scaling and the true optimal dynamic policy. We focus on a special class of input processes: Poisson arrivals and a degenerate Hyperexponential job size distribution (a mix of a point mass at 0 and an Exponential distribution). This allows us to compute optimal dynamic policies using the algorithm proposed by [17]. The dynamic policy for the diffusion control



problem was computed using the Newton-Raphson method (Algorithm 2, Section 4.4). The service rate curve is the one shown in Figure 1, which gives the drift function as

$$-\theta(x) \doteq \lambda m \log \frac{\lambda m}{\hat{\mu}(x)} \doteq \lambda m \log \frac{\lambda m}{1.25 - \frac{x^2}{150}}. \quad (32)$$

We used MATLAB's `ode45` function to solve the differential equations involved in Algorithm 2. The performance of the diffusion control policy was evaluated by simulating it for a Poisson arrival process and Hyperexponential job size distribution with the indicated  $c_s^2$ .

For each of the six cases shown, the steady-state mean number of jobs for the diffusion control heuristic is within 2% of the optimal dynamic policy, demonstrating the validity of our proposed scaling for computing control policies for Sd-LPS systems across a range of traffic intensities.

|              |                 | Steady-state mean number of jobs $\mathbf{E}[N]$ |                          |                   |
|--------------|-----------------|--|--------------------------|-------------------|
|              |                 | Opt. dynamic policy                              | Diffusion control policy | Suboptimality (%) |
| $c_s^2 = 4$  | $\lambda = 0.7$ | 1.739  | 1.744                    | 0.29              |
|              | $\lambda = 0.8$ | 2.854  | 2.885                    | 1.09              |
|              | $\lambda = 0.9$ | 4.873  | 4.893                    | 0.41              |
| $c_s^2 = 19$ | $\lambda = 0.7$ | 2.90   | 2.94                     | 1.38              |
|              | $\lambda = 0.8$ | 6.51   | 6.63                     | 1.84              |
|              | $\lambda = 0.9$ | 14.24  | 14.33                    | 1.01              |

Table 1: Simulation results comparing the performance of dynamic policies for Poisson arrivals with rate  $\lambda$  and a degenerate hyperexponential ( $H^*$ ) job size distribution with  $m = 1$  and SCV  $c_s^2$ . The first column shows the steady-state mean number of jobs for the optimal dynamic policy. The second column shows the same metric for the heuristic policy obtained from the diffusion control problem. For each case, the diffusion control policy yields an  $\mathbf{E}[N]$  of at most 2% larger than the optimal policy.

## 4.2 The diffusion control problem

In this section we set up the diffusion control problem for dynamic concurrency control of the LPS server. We begin by generalizing the scaling of the concurrency limit  $k^{(r)}$  given in (9) so that it becomes a function of the workload in the system:

$$k^{(r)}(W^{(r)}(t)) = rk \left( \frac{W^{(r)}(t)}{r} \right), \quad (33)$$

where  $k : \mathbb{R}_+ \rightarrow \mathbb{R}_+$  and  $k(w) \leq w/m_e$  for any  $w$ . The restriction  $k(w) \leq w/m_e$  on the choice of concurrency level is driven by the state space collapse (see Conjecture 1).

The objective is to find the optimal state-dependent concurrency level function  $k(\cdot)$ . For technical reasons, we restrict our consideration to the following family of dynamic controls

$$\mathcal{K} = \left\{ k : \mathbb{R}_+ \rightarrow \mathbb{R}_+ \mid k(w) \leq w/m_e; \ k \text{ is Lipschitz continuous; } \int_{v=0}^{\infty} e^{-\int_0^v \theta(k(w))dw} dv < \infty \right\}. \quad (34)$$

The Lipschitz continuity requirement is for technical reasons, and the last condition above is only to ensure that a stationary distribution for the diffusion-scaled workload under  $k(w)$  exists. We use

the same heavy traffic regime as in Section 3.2 except that stability condition (17) is replaced by

$$\sup_{x \in [0, M]} \theta(x) > 0, \quad (35)$$

for some  $M < \infty$ . That is, a service rate strictly larger than the arrival rate is achievable at a finite concurrency limit and hence at a finite workload. In fact, we will make a stronger assumption on  $\sup_x \theta(x)$ . Define

$$\hat{\theta} \doteq \sup_{x \in \mathbb{R}_+} \theta(x); \quad \hat{k} \doteq \arg \max_k \{\theta(k)\}.$$

Here  $\hat{k}$  denotes the most efficient concurrency level, which we will assume to be finite. For any  $k \in \mathcal{K}$ , define the mapping  $\Delta_k : \mathbb{R}_+ \rightarrow \mathbb{R}_+$  by

$$\Delta_k(w) = \frac{w \wedge k(w)m_e}{m_e} + \frac{(w - k(w)m_e)^+}{m}. \quad (36)$$

Note that we use  $\Delta_K$  to denote the mapping under a static concurrency level  $K$ , and  $\Delta_k$  to denote the mapping under a dynamic concurrency policy  $k \in \mathcal{K}$ . Extending the diffusion limit result in Section 3.2, we have the following conjecture:

**Conjecture 1 (Diffusion limits under a dynamic policy)** *For the sequence of Sd-LPS systems parametrized by  $r \in \mathbb{Z}^+$  under the dynamic policy (33) for some  $k \in \mathcal{K}$ , as  $r \rightarrow \infty$ ,*

$$\hat{W}^{(r)} \Rightarrow W^*, \quad (37)$$

*where  $W^*$  is an RBM with initial value  $W^*(0) = w^*$ , drift  $-\theta(\Delta_K(W^*(t)) \wedge k(W^*(t)))$  and variance  $\sigma^2 = \lambda m^2(c_a^2 + c_s^2)$ . Moreover, as  $r \rightarrow \infty$ ,*

$$\hat{X}^{(r)} \Rightarrow X^* = \Delta_k(W^*). \quad (38)$$

In other words, we conjecture that the state space collapse result still holds and the state-dependent concurrency level function  $k(\cdot)$  only plays a role in modifying the drift of the diffusion limit of the workload. The key to proving this conjecture is to extend the state space collapse result to allow a dynamic concurrency level and analyze the underlying fluid model (as in [45]). Due to the technical intricacies involved, proving the conjecture is beyond the scope of this paper. Instead, we focus on utilizing the conjectured diffusion limit to identify a near-optimal policy for the original LPS system.

As mentioned earlier, we will formulate the diffusion control problem with the limiting workload process  $W^*$  as the state variable and the map  $\Delta_k(\cdot)$  as the state-dependent cost function (using the state space collapse conjecture (38)). There are two reasons for choosing  $W^*$  over the head count process  $X^*$  as the state variable: (i) the variance of  $X^*$  is state-dependent making the computation more complicated, while it is a constant for  $W^*$ , and (ii) headcount does not carry enough information since two different states  $(Q_1, Z_1)$  and  $(Q_2, Z_2)$  along the state-space collapse trajectory may have the same head count but different workloads. Therefore the control is not uniquely obtained as a function of the number of jobs in the system.

Let  $V_\gamma(w)$  denote the discounted total cost (with discount rate  $\gamma$ ) for the limiting process  $W^*$  under a control policy  $k(\cdot)$  when the workload starts in state  $w$ :

$$V_\gamma(w) = \mathbb{E}_w \left[ \int_0^\infty e^{-\gamma t} \Delta_k(W^*(t)) dt \right]. \quad (39)$$

## Optimality Equations

Consider a small  $\delta > 0$ . According to Itô calculus

$$\begin{aligned} V_\gamma(w) &= \Delta_k(w) + (1 - \gamma\delta)\mathbb{E}[V_\gamma(W^*(\delta))] + o(\delta) \\ &= \Delta_k(w) + (1 - \gamma\delta)\mathbb{E}\left[V_\gamma(w) + V'_\gamma(w)(W^*(\delta) - w) + \frac{V''_\gamma(w)}{2}(W^*(\delta) - w)^2 + o(\delta)\right] + o(\delta) \\ &= \Delta_k(w) + (1 - \gamma\delta)\left[V_\gamma(w) + V'_\gamma(w)\theta(k(w))\delta + \frac{V''_\gamma(w)}{2}\sigma^2\delta\right] + o(\delta), \end{aligned}$$

where, recall,  $\sigma^2 \doteq \lambda m^2(c_s^2 + c_a^2)$ . We thus have the following relation for the discounted value function  $V_\gamma$ :

$$\gamma V_\gamma(w) = \Delta_k(w) - \theta(k(w))V'_\gamma(w) + \frac{\sigma^2}{2}V''_\gamma(w). \quad (40)$$

Letting  $\gamma \rightarrow 0$ , define

$$v = \lim_{\gamma \rightarrow 0} \gamma V_\gamma(w), \quad \text{and} \quad G(w) = \lim_{\gamma \rightarrow 0} V'_\gamma(w),$$

where  $v$  is the average cost of policy  $k(\cdot)$ , and the value function gradient  $G(w)$  solves the following ordinary differential equation (ODE):

$$v = \Delta_k(w) - \theta(k(w))G(w) + \frac{\sigma^2}{2}G'(w). \quad (41)$$

Above, we have provided a heuristic derivation to arrive at the average cost optimal control problem as a limit of the discounted cost problem. For a formal treatment of the relation between discounted and average cost problems (i.e., by defining discounted relative cost functions  $h_\gamma(w) = V_\gamma(w) - V_\gamma(\tilde{w})$  for some positive recurrent state  $\tilde{w}$ , taking limit  $h(w) = \lim_{\gamma \downarrow 0} h_\gamma(w)$  and  $v = \lim_{\gamma \downarrow 0} \gamma V_\gamma(\tilde{w})$ ), we refer readers to [22, Chapter 5], [6].

**Proposition 3** *The discounted value function  $V_\gamma(w)$  is non-decreasing in  $w$  for all  $\gamma$ , and hence  $G(w) \geq 0$ .*

**Remark 1** *For a given control policy  $k(w)$ , equation (41) is a first order ODE for  $G(w)$ . However, to solve  $G(\cdot)$  we also need to know the average cost  $v$ . This is to be expected since we started from a second order ODE where we would need two boundary conditions to completely specify  $V_\gamma$ . In our case, one boundary condition is easy to get hold of: since we have a reflecting boundary at  $w = 0$ , we must have (see, for example, [33, page VIII]):*

$$V'_\gamma(0) = 0 \quad (42)$$

*and therefore, also  $G(0) = 0$ . This observation will be critical in the development of our algorithms.*

Returning to equation (40), let  $V_\gamma^*$  denote the value function for the optimal policy. Then Bellman's principle of optimality becomes:

$$\gamma V_\gamma^*(w) = \min_{k \in [0, w/m_e]} \left\{ \Delta_k(w) - \theta(k)V_\gamma^{*'}(w) + \frac{\sigma^2}{2}V_\gamma^{*''}(w) \right\}. \quad (43)$$

If we let  $\gamma \rightarrow 0$ , then

$$v^* = \min_{k \in [0, w/m_e]} \left\{ \Delta_k(w) - \theta(k)G^*(w) + \frac{\sigma^2}{2}G^{*'}(w) \right\}, \quad (44)$$

where again, as remarked earlier, we have the boundary condition  $G^*(0) = 0$ , leaving  $v^*$  the only unknown.

Though many diffusion control problems addressed in the literature have a nice structure allowing a closed-form solution, e.g., [19, 20], the problem (44) is intrinsically difficult mainly due to the generality of the service rate curve. Thus we seek numerical algorithms, which presents another challenge. For diffusion control problems where a closed-form solution can be found, one of the boundary conditions is imposed by setting the coefficient of the exponential term in the solution of the second order ODE to zero. This captures the physical constraint that the optimal value function should asymptotically grow at a polynomial rate and not exponentially. However, this trick cannot be applied when searching for a numerical solution, which led us to develop the algorithms in Sections 4.3 and 4.4 to get around this obstacle. While the majority of numerical algorithms for solving diffusion control problems rely on the Markov chain method where time and space are discretized and a probability transition matrix is engineered to satisfy local consistency requirements (e.g., [28]), we directly work with the ODE in (44).

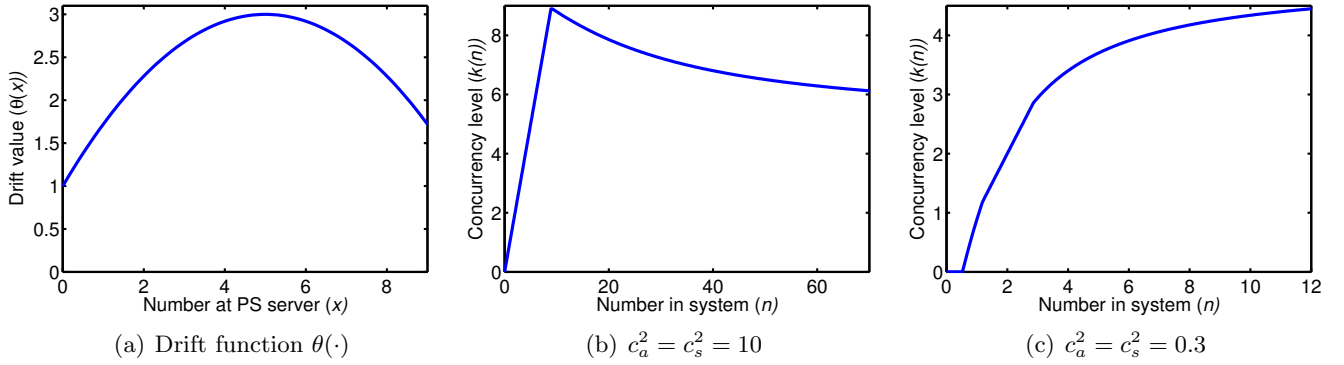


Figure 3: A hypothetical drift function  $\theta(x)$  and the optimal diffusion control policies for two choices of workload parameters  $c_a^2, c_s^2$ .

For an illustration of what an optimal dynamic policy might look like, see Figure 3. The first figure shows an illustrative example of the  $\theta(x)$  function for the PS server. As can be seen, the PS server is most efficient when there are  $\hat{k} = 5$  jobs at the server, and the speed drops on either side of this point. The second figure shows the optimal dynamic policy (translated from  $k(w)$  to  $k(n)$ , that is, as a function of the number of jobs in the system, for clarity) when  $c_s^2 = c_a^2 = 10$ . This corresponds to a workload that has significant variability, and the optimal policy increases the concurrency level to approximately 9 when the number of jobs in the system is small but scales it back when there is a long queue. The third figure shows the policy for  $c_s^2 = c_a^2 = 0.3$ . This is a low variability workload, and as the number of jobs in the system increases, initially the PS server acts as an FCFS server and thus compromises speed to keep the concurrency level small. At  $n \approx 0.5$ , the system switches to a controlled PS behavior by gradually increasing the concurrency level to increase service rate. At  $n \approx 1.2$  the system switches to a pure PS behavior admitting everyone in queue, and finally at  $n = 3$  it switches back to a controlled PS behavior, gradually increasing the concurrency level to  $\hat{k} = 5$  as queue becomes longer. The graphs shown were produced using the Newton-Raphson average cost iteration algorithm described in Section 4.4.

### 4.3 Binary search algorithm for solving the diffusion control problem

Before giving the algorithm, we discuss the main intuition and ideas behind it. Let us assume that an oracle reveals to us the average cost  $v^*$  of the optimal policy. This, together with the boundary condition  $G^*(0) = 0$ , would allow us to numerically solve for the optimal control by evolving  $G^*(\cdot)$  forward: Assuming we have solved  $G^*(w)$  for  $w \in [0, x]$ , we first find

$$k^*(x) = \arg \min_{k \in [0, x/m_e]} \left\{ k \left( 1 - \frac{m_e}{m} \right) - \theta(k) G^*(x) \right\} \quad (45)$$

and then

$$\frac{\sigma^2}{2} G^{*'}(x) = v^* - \left[ \frac{x}{m} + k^*(x) \left( 1 - \frac{m_e}{m} \right) - \theta(k^*(x)) G^*(x) \right]$$

allows us to evolve  $G^*(w)$  forward in a small enough interval  $(x, x + \delta x]$ . Here then is the idea of the binary search algorithm in a nutshell: we maintain an interval  $[L, U]$  within which  $v^*$  is known to lie. We test if  $\frac{L+U}{2}$  is the average cost of a *feasible* control (we describe the feasibility test shortly). If it is, then  $v^*$  is at most  $\frac{L+U}{2}$  and we update  $U$  to this value, otherwise we update  $L$  to this value. Therefore, within  $O(\log \frac{1}{\epsilon})$  iterations, we would have  $(U - L) \leq \epsilon$ , at which point we return a control corresponding to the average cost  $U$  (which is feasible). (If we were interested in solving the discounted cost problem, the only thing to change would be to search the value of  $\gamma V_\gamma^*(0)$  instead of the average cost.)

**Detecting infeasibility ( $v < v^*$ ) :** Let us assume that we guess a value  $v$  that is smaller than the optimal average cost  $v^*$  and solve the ODE (44) forward starting from 0. What would go wrong? It turns out that in this case  $G^*(w) < 0$  for some  $w > 0$ , contradicting Proposition 3. Indeed, while  $v < v^*$  is infeasible for the original LPS system, it is still the cost of a feasible policy for a *finite buffer LPS loss system*. If  $\underline{W}(v)$  denotes the smallest  $w$  at which  $G^*(w) < 0$ , then  $v$  is the optimal cost of the finite buffer system with buffer size  $\underline{W}(v)$ , and during the forward evolution of  $G^*(\cdot)$  we have in fact found the optimal control policy for this finite buffer loss system. This provides us with a one-sided test of infeasibility: if ever  $G^*(w) < 0$  for some  $w > 0$ , our guess  $v$  is too optimistic (i.e.,  $v < v^*$ ). We formalize these statements in the proposition below:

**Proposition 4** *Let  $v^*$  be the average cost of the optimal control for the diffusion control problem (44), and let  $v < v^*$ . Let  $G_v(\cdot)$  be the solution of the Bellman equation:*

$$v = \min_{k \in [0, w/m_e]} \left[ \frac{w}{m} + k \left( 1 - \frac{m_e}{m} \right) - \theta(k) G_v(w) \right] + \frac{\sigma^2}{2} G'_v(w) \quad (46)$$

*with initial condition  $G_v(0) = 0$ , and  $k_v^*(w)$  be the policy obtained while solving for  $G_v(\cdot)$ . Then  $k_v^*(w)$  is the optimal control policy for a finite (workload) buffer system with buffer limit  $\underline{W}(v)$ , where*

$$\underline{W}(v) = \inf \{ w > 0 : G_v(w) < 0 \}. \quad (47)$$

*Further,  $\underline{W}(v) = O \left( \log \frac{1}{v^* - v} \right)$ .*

Since we do not have a priori a concrete bound on the value of  $w$  for which  $G_v(w) < 0$  for  $v < v^*$ , this infeasibility test alone cannot be translated into an efficient algorithm to find  $v^*$ . We therefore need a test of feasibility, which is provided next.

**Detecting feasibility ( $v > v^*$ ) :** Detecting infeasibility relied on (a) choosing a set of alternate systems (finite buffer LPS loss systems) which lower bound the cost of the original problem, but have the same HJB equation as the original system albeit with a different boundary condition ( $G(W) = 0$ ), and (b) being able to identify in bounded time which finite buffer system our guess  $v$  maps to through satisfaction of a second boundary condition (although we only expressed this bound in order notation instead of a concrete bound).

To test if a guess  $v$  is larger than the true optimal cost  $v^*$ , we will employ a class of suboptimal policies we call *fluid continuation policies*.

**Definition 1** *The set of fluid continuation policies with fluid continuation point  $W$  is defined as*

$$\mathcal{F}_W = \left\{ k \in \mathcal{K} : k(w) = k_f(w) \doteq \arg \max_{x \leq w/m_e} \{\theta(x)\}, w \geq W \right\}. \quad (48)$$

That is, beyond the fluid continuation workload point  $W$ , the control is chosen to be the most efficient service rate available. Denote the cost of the optimal (minimum cost) policy in  $\mathcal{F}_W$  by  $v_f(W)$ . Let  $\bar{W}(v) = \min\{W \geq 0 : v = v_f(W)\}$ .

Clearly all policies in  $\mathcal{F}_W$  are stable due to condition (35). In fact, the policy  $k_f \in \mathcal{F}_0$  is optimal when  $c_s^2 = 1$  since in this case  $m_e = m$ , and (45) simplifies to

$$k^*(w) = \arg \min_{k \in [0, w/m_e]} \theta(k) G^*(w) = \arg \min_{k \in [0, w/m_e]} \theta(k).$$

We will use the next proposition to answer the question whether or not any given guess of the average cost is feasible.

**Proposition 5** *Let  $v^* \leq v \leq v_f(0)$ . Then  $v$  is the average cost of an optimal fluid continuation policy  $k_v$  with continuation point  $\bar{W}(v)$ . That is:*

$$k_v(w) = \begin{cases} \arg \min_{k \in [0, w/m_e]} \{k(1 - \frac{m_e}{m}) - \theta(k)G_v(w)\} & w \leq \bar{W}(v), \\ k_f(w) & w > \bar{W}(v), \end{cases} \quad (49)$$

where  $G_v$  is the value function gradient for policy  $k_v$  and satisfies the ODE

$$v = \frac{w}{m} + k_v(w) \left(1 - \frac{m_e}{m}\right) - \theta(k_v(w))G_v(w) + \frac{\sigma^2}{2}G_v'(w). \quad (50)$$

Further,  $\bar{W}(v) = O\left(\log \frac{1}{v-v^*}\right)$ .

The advantage of the class of feasible policies described by (49) is that for any  $v$ , the function  $G_v(w)$  for  $w \geq \bar{W}(v)$  can be easily computed, and is in fact independent of  $\bar{W}(v)$ . Let us call this the *fluid continuation of the value function gradient* and denote it by  $\bar{G}_v(w)$ . The function  $\bar{G}_v$  will act as the boundary condition for detecting the feasibility of  $v$ .

$\bar{G}_v(w)$  solves the ODE

$$v = \frac{w}{m} + k_f(w) \left(1 - \frac{m_e}{m}\right) - \theta(k_f(w))\bar{G}_v(w) + \frac{\sigma^2}{2}\bar{G}_v'(w) \quad (51)$$

with an as-yet-unspecified boundary condition. Note that  $\overline{G}_v(0)$  is not necessarily 0, unless  $v$  happens to be the average cost of policy  $k_f$  (that is,  $v = v_f(0)$ ). We first consider the range  $w \in [\hat{k}m_e, \infty)$  where  $k_f(w) = \hat{k}$ . In this part, the ODE for  $\overline{G}_v$  becomes

$$v = \frac{w}{m} + \hat{k} \left(1 - \frac{m_e}{m}\right) - \hat{\theta} \overline{G}_v(w) + \frac{\sigma^2}{2} \overline{G}_v'(w)m, \quad (52)$$

which is a first order non-homogeneous differential equation with constant coefficients, and a non-zero part that is a linear function of  $w$ . The solution to (51) has a homogeneous (general) part and a particular solution with two unknown constants. To determine the unknown constants we need two boundary conditions, one of which is  $\overline{G}_v(0) = 0$ . We obtain our equivalent of the second boundary condition by setting the coefficient of the homogeneous part of the solution (which is  $e^{\frac{2\hat{\theta}w}{\sigma^2}}$ ) to zero because the value function cannot grow exponentially. This gives

$$\overline{G}_v(w) = \frac{w}{m\hat{\theta}} + \left( \hat{k} \left(1 - \frac{m_e}{m}\right) + \frac{\sigma^2}{2m\hat{\theta}} - v \right) \frac{1}{\hat{\theta}}, \quad w \geq \hat{k}m_e. \quad (53)$$

To solve for  $\overline{G}_v$  for  $w \in [0, \hat{k}m_e]$ , we solve the ODE (51) backwards starting with the terminal condition

$$\overline{G}_v(\hat{k}m_e) = \left( \hat{k} - v + \frac{\sigma^2}{2m\hat{\theta}} \right) \frac{1}{\hat{\theta}}.$$

If it turns out that  $\overline{G}_v(0) < 0$ , then our guess  $v > v_f(0)$  and therefore  $v$  is the average cost of some feasible policy. Otherwise, we solve the ODE (44) forward by starting with initial condition  $G_v(0) = 0$  and substituting our guess  $v^* = v$ . If  $G_v$  ‘hits’  $\overline{G}_v$  from below, that is  $G_v(W) = \overline{G}_v(W)$  for some  $W$ , then  $\overline{W}(v) = W$ , and following policy  $k_v(w)$  for  $w \in [0, \overline{W}(v)]$  and  $k_f$  for  $w \geq \overline{W}(v)$  is a policy with average cost  $v$ , indicating feasibility of  $v$ .

The step-by-step procedure is given in Algorithm 1.

#### 4.4 Newton-Raphson method for solving the diffusion control problem

The binary search algorithm we proposed in Section 4.3 was based on first guessing an average cost value  $v$ , forward evolving ODE (44) with the (initial) boundary condition  $G_v(0) = 0$  until a terminal boundary condition was met thereby verifying feasibility or infeasibility of  $v$  as the average cost, and then updating the guess for  $v$ . The algorithm we propose in this section will be based on backward evolution of ODE (44).

As in Section 4.3, here we will find an optimal policy in the class of fluid continuation policies (Definition 1)  $\mathcal{F}_W$  for some fluid continuation point  $W$ . However, this time we will first fix a large enough value of  $W$  ( $W \geq \hat{k}m_e$ ) and seek the optimal policy and the optimal average cost in  $\mathcal{F}_W$ . Recall that we denote the average cost of the optimal policy in  $\mathcal{F}_W$  by  $v_f(W)$ . As mentioned later, we can use a standard doubling trick to settle on a ‘large enough’  $W$ . Next, we will guess an average cost value  $v$  and devise a test to compare  $v$  with  $v_f(W)$ . For this, we evolve ODE (44) backwards with the (terminal) boundary condition

$$G_v(W) = \left( \hat{k} \left(1 - \frac{m_e}{m}\right) - v + \frac{\sigma^2}{2m\hat{\theta}} \right) \frac{1}{\hat{\theta}} + \frac{1}{m\hat{\theta}} \cdot W \quad (54)$$

(For notational simplicity, we have suppressed the dependence of  $G_v()$  on  $W$ ). If indeed  $v = v_f(W)$  then we must have  $G_v(0) = 0$ , and the sign of  $G_v(0)$  can tell us if  $v < v_f(W)$  or  $v > v_f(W)$ . This



---

**Algorithm 1** Average cost iteration (binary search method)

---

**define**  $\hat{k} \doteq \arg \max_k \theta(k)$ ;  $\hat{\theta} \doteq \theta(\hat{k})$   
**define**  $k_f(w) \doteq \arg \max_{k \in [0, w/m_e]} \theta(k)$ ;  $\theta_f(w) \doteq \theta(k_f(w))$  ▷ (Fluid optimal policy)  
**initialize**  $L \leftarrow 0, U \leftarrow \emptyset, v \leftarrow 1$  ▷ (Search interval  $[L, U]$  and initial guess)  
**while**  $U - L \geq \epsilon$  **do**  
    **solve** for the fluid continuation  $\overline{G}_v(w)$  with average cost  $v$ :  
         $\overline{G}_v(w) = \frac{w}{m\hat{\theta}} + \left( \hat{k} \left(1 - \frac{m_e}{m}\right) + \frac{\sigma^2}{2m\hat{\theta}} - v \right) \frac{1}{\hat{\theta}}$  ▷ (Terminal condition)  $\dots w \in [\hat{k}m_e, \infty)$   
         $v = \frac{w}{m} + k_f(w) \left(1 - \frac{m_e}{m}\right) - \theta_f(w) \overline{G}_v(w) + \frac{\sigma^2}{2} \overline{G}'_v(w)$  ▷ (ODE)  $\dots w \in [0, \hat{k}m_e]$   
    **end solve**  
    **if**  $\overline{G}_v(0) < 0$  **then** ▷ ( $v$  is larger than avg. cost of  $k_f$ )  
         $U \leftarrow v$  ▷ (Therefore,  $v^* \leq v$ )  
         $v \leftarrow \frac{L+U}{2}$  ▷ (Updated guess for next iteration)  
    **else**  
        **solve** for policy  $k_v(w)$  and  $G_v(w)$ :  
             $G_v(0) = 0$  ▷ (Initial condition)  
             $v = \min_{k \in [0, w/m_e]} \left\{ \frac{w}{m} + k \left(1 - \frac{m_e}{m}\right) - \theta(k) G_v(w) + \frac{\sigma^2}{2} G'_v(w) \right\}$  ▷ (ODE)  
            **until**  $W = \inf\{w : (G_v(w) \geq \overline{G}_v(w)) \text{ OR } (G_v(w) < -\epsilon)\}$  ▷ (Terminal event)  
        **end solve**  
        **if**  $G_v(W) \geq \overline{G}_v(W)$  **then** ▷ ( $v$  is feasible)  
             $U \leftarrow v$  ▷ (Therefore,  $v^* \leq v$ )  
             $v \leftarrow \frac{L+U}{2}$  ▷ (Update guess for next iteration)  
        **else** ▷ ( $v$  is infeasible)  
            **if**  $U = \emptyset$  **then**  
                 $v \leftarrow 2v$  ▷ (Double the guess until we find one feasible value)  
            **else**  
                 $L \leftarrow v$  ▷ ( $v^* \geq v$ )  
                 $v \leftarrow \frac{L+U}{2}$  ▷ (Update guess for next iteration)  
            **end if**  
        **end if**  
    **end while**  
**return** Cost  $v = U$ ; Policy  $k_U(w)$

---

would be similar to the binary search iterative algorithm with a linear convergence rate. However, since we know  $G_{v_f(W)}(0) = 0$ , we can (and will) find  $v_f(W)$  by solving for the root of the equation  $G_v(0) = 0$  (in  $v$ ) using the Newton-Raphson method which has a faster quadratic convergence rate.

Recall that to solve for the root of a function  $h(x) = 0$  with a current estimate of  $x_n$ , the Newton-Raphson update step is given by

$$x_{n+1} = x_n - \frac{h(x_n)}{h'(x_n)}.$$

Let us assume that our current guess for  $v_f(W)$  is  $v_n$ . To generate the next guess  $v_{n+1}$  via the Newton-Raphson method, we need the derivative of  $G_v(0)$  at  $v = v_n$ . With some abuse of notation, define

$$g_v(w) \doteq \frac{dG_v(w)}{dv}.$$

(What we really mean by the above is that  $g_v(w) \doteq \frac{\partial G(v,w)}{\partial v}$ , where  $G(v,w) = G_v(w)$ .) As we will show in the proof of Proposition 6 (see step 2 of the proof),  $G_v(w)$  is Lipschitz continuous and decreasing in  $v$  for all  $w$  and therefore  $g_v(w)$  exists almost everywhere, and further it is bounded away from 0. With  $W \geq \hat{k}m_e$  representing the point at which we switch to the fluid policy  $k_f$ , we can write  $G_v(w)$  as the following integral: for  $w \leq W$ ,

$$G_{v_n}(w) = G_{v_n}(W) + \frac{2}{\sigma^2} \int_W^w \left[ v_n - \min_{k \in [0, u/m_e]} \{ \Delta_k(u) - \theta(k)G_{v_n}(u) \} \right] du, \quad (55)$$

where  $G_{v_n}(W)$  is given by (54). Differentiating (55) with respect to  $v_n$  yields

$$\begin{aligned} g_{v_n}(w) &= g_{v_n}(W) + \frac{2}{\sigma^2} \int_W^w \left[ 1 - \frac{d}{dv_n} \min_{k \in [0, u/m_e]} \{ \Delta_k(u) - \theta(k)G_{v_n}(u) \} \right] du \\ &= -\frac{1}{\hat{\theta}} + \frac{2}{\sigma^2} \int_W^w [1 + \theta(k_{v_n}(u))g_{v_n}(u)] du. \end{aligned}$$

Since the policy  $k_{v_n}()$  also depends on  $v_n$ , to arrive at the last equality, we have used the *envelope theorem*: If  $k^*(v) = \arg \min_k \phi(k, v)$  and  $\phi^*(v) = \phi(k^*(v), v)$ , then  $\frac{d\phi^*(v)}{dv} = \frac{\partial \phi(k^*(v), v)}{\partial v}$  (where  $\frac{\partial \phi(k, v)}{\partial v}$  is the partial derivative with respect to  $v$ ). Therefore, very similar to  $G_{v_n}$ ,  $g_{v_n}$  satisfies the following ODE

$$1 = -\theta(k_{v_n}(w))g_{v_n}(w) + \frac{\sigma^2}{2} g'_{v_n}(w) \quad (56)$$

with the terminal condition

$$g_{v_n}(W) = -\frac{1}{\hat{\theta}}. \quad (57)$$

The updated guess for average cost is then

$$v_{n+1} = v_n - \frac{G_{v_n}(0)}{g_{v_n}(0)}. \quad (58)$$

Remarkably, it turns out that  $v_{n+1}$  is exactly the average cost of the policy, call it  $k_{v_n}(w)$ , that is implicitly generated when solving for  $G_{v_n}$  and  $g_{v_n}$ . This is because if we fix the policy  $k(w) = k_{v_n}(w)$ , then from (55) we can see that  $G_v$  is linear in  $v$ . Therefore the Newton-Raphson update is effectively solving for that  $v$  for which  $G_v(0) = 0$  when  $k(w) = k_{v_n}(w)$ , which is the average cost of  $k_{v_n}$ . Therefore the sequence of average cost iterates produced by the algorithm are in fact the average costs of a sequence of feasible policies. The next proposition formally states the result on convergence of the Newton-Raphson average cost iteration algorithm.

**Proposition 6** *Let  $v_1, v_2, \dots$  denote the average cost iterates generated by the Newton-Raphson method (58). Let*

$$d_\theta \doteq \sup_k \theta(k) - \inf_k \theta(k) < \infty.$$

*The sequence  $\{v_n\}$  monotonically decreases to  $v_f(W)$ , which is the average cost of the optimal diffusion control policy in the set  $\mathcal{F}_W$  of fluid continuation policies with fluid continuation point  $W$ . Further, assuming that  $\theta(k)$  is twice differentiable everywhere, and that*

1. *the first derivative of  $\theta(k)$  is finite, i.e.,*

$$S_\theta \doteq \sup_k \left| \frac{d\theta(k)}{dk} \right| < \infty, \quad (59)$$

2. *the second derivative of  $\theta(k)$  is bounded away from 0, i.e.,*

$$D_\theta \doteq \inf_k \left| \frac{d^2\theta(k)}{dk^2} \right| > 0, \quad (60)$$

*the errors of the Newton-Raphson iterates,  $\epsilon_n = (v_n - v_f(W))$ , decrease quadratically.*

Since close to the root, the error roughly squares in each iteration, it takes  $O(\log \log \frac{1}{\epsilon})$  iterations to reach an  $\epsilon$ -optimal policy within  $\mathcal{F}_W$ . To find the  $\epsilon$ -optimal policy among all policies, we can keep doubling the value of  $W$  until the error between successive iterates is sufficiently small. By our earlier result, we need a  $W = O(\log \frac{1}{\epsilon})$  to arrive at an  $\epsilon$ -optimal policy. Since each iteration of the Newton-Raphson method takes  $O(W)$  time, the overall time complexity of the algorithm to find an  $\epsilon$ -optimal policy is  $O(\log \frac{1}{\epsilon} \log \log \frac{1}{\epsilon})$ . On the other hand, the overall time complexity of the binary search algorithm is  $O\left(\left(\log \frac{1}{\epsilon}\right)^2\right)$ .

The step-by-step procedure is described in Algorithm 2. In the description, we have omitted iterating over values of  $W$ , the fluid continuation point, for clarity and to focus on the core of the algorithm. Note that we choose the average cost of the fluid policy as the initial guess for average cost  $v_0$  which is computed using a single step of Newton-Raphson iteration (shown in the initialize block). This is a minor optimization that also takes care of a corner case in the proof of convergence of Algorithm 2, although we believe this initialization is not necessary for quadratic convergence to hold.

**Comparison with the policy iteration algorithm:** Puterman and Brumelle [35] formally proved that the policy iteration algorithm for discrete-time Markov decision processes is equivalent to the Newton-Raphson algorithm for finding the fixed point of the dynamic programming functional equation, but performed in the value function space. Puterman [36] presented a policy iteration algorithm for control of a diffusion process in a bounded region in  $\mathbb{R}^n$  for finite horizon total cost optimization. It is therefore instructive to compare our average cost iteration algorithm with his policy iteration algorithm for control of diffusions. One difference is that we carry out the Newton-Raphson algorithm in the space of average cost. Another major difference is that the policy iteration algorithm alternates between policy evaluation and policy improvement steps. Our algorithm can be viewed as one where we have folded the policy evaluation and policy improvement into one step.

---

**Algorithm 2** Average cost iteration (Newton-Raphson method)

---

**define**  $\hat{k} \doteq \arg \max_k \theta(k)$ ;  $\hat{\theta} \doteq \theta(\hat{k})$   
**require**  $W \geq \hat{k}m_e$  ▷ (Fluid continuation point)  
**initialize** ▷ (Compute initial guess for average cost  $v_f(0)$ , see Defn. 1)  
    **solve** functions  $G_f(w)$  and  $g_f(w)$  for  $w \in [0, \hat{k}m_e]$ :  
         $G_f(\hat{k}m_e) = \left(\hat{k}(1 - m_e/m) + \frac{\sigma^2}{2m\hat{\theta}}\right) \frac{1}{\hat{\theta}} + \frac{1}{m\hat{\theta}} \cdot \hat{k}m_e$  ▷ (Terminal condition for  $G_f$ )  
         $g_f(\hat{k}m_e) = -\frac{1}{\hat{\theta}}$  ▷ (Terminal condition for  $g_f$ )  
         $k_f(w) = \arg \max_{k \in [0, w/m_e]} \theta(k)$  ▷ (Fluid optimal policy)  
         $0 = \frac{w}{m} + k_f(w)(1 - \frac{m_e}{m}) - \theta(k_f(w))G_f(w) + \frac{\sigma^2}{2}G'_f(w)$  ▷ (ODE for  $G_f$ )  
         $1 = -\theta(k_f(w))g_f(w) + \frac{\sigma^2}{2}g'_f(w)$  ▷ (ODE for  $g_f$ )  
    **end solve**  
     $v_0 \leftarrow v_f(0) = -\frac{G_f(0)}{g_f(0)}$   
**end initialize**  
**repeat**  
    **solve** policy  $k_{v_n}(w)$ , functions  $G_{v_n}(w)$  and  $g_{v_n}(w)$  for  $w \in [0, W]$ :  
         $G_{v_n}(W) = \left(\hat{k}(1 - m_e/m) - v_n + \frac{\sigma^2}{2m\hat{\theta}}\right) \frac{1}{\hat{\theta}} + \frac{1}{m\hat{\theta}} \cdot W$  ▷ (Terminal condition for  $G_{v_n}$ )  
         $g_{v_n}(W) = -\frac{1}{\hat{\theta}}$  ▷ (Terminal condition for  $g_{v_n}$ )  
         $k_{v_n}(w) = \arg \min_{k \in [0, w/m_e]} \left\{ k \left(1 - \frac{m_e}{m}\right) - \theta(k)G_{v_n}(w) \right\}$   
         $v_n = \frac{w}{m} + k_{v_n}(w)(1 - \frac{m_e}{m}) - \theta(k_{v_n}(w))G_{v_n}(w) + \frac{\sigma^2}{2}G'_{v_n}(w)$  ▷ (ODE for  $G_{v_n}$ )  
         $1 = -\theta(k_{v_n}(w))g_{v_n}(w) + \frac{\sigma^2}{2}g'_{v_n}(w)$  ▷ (ODE for  $g_{v_n}$ )  
    **end solve**  
    **update**  $v_{n+1} \leftarrow v_n - \frac{G_{v_n}(0)}{g_{v_n}(0)}$  ▷ (Newton-Raphson update)  
**until**  $|G_{v_n}(0)| \leq \epsilon$   
**return** Cost  $v_{n+1}$ ; Policy  $k_{v_n}(w)$

---

## 5 Concluding Remarks

The primary goal of the present paper was to propose a diffusion scaling to aid the analysis and control of State-dependent Limited Processor Sharing (LPS) systems. Our philosophy while designing the scaling was to fix a limiting distribution for the steady-state number of jobs in the system, and then reverse-engineer the sequence of service rate curves that yields this limit. By choosing the limiting distribution as the one of the original state-dependent system under an  $M/M/$  input, our scaling captures the effect of the entire service rate curve. The resulting diffusion approximation leads to the choice of a near-optimal static concurrency limit.

To compute dynamic control policies, we generalized our scaling by defining it directly in terms of the service rate curves of the original system. Again, as proof-of-concept, we presented experimental results demonstrating that the dynamic policies resulting from the associated diffusion control problem are extremely close to the optimal dynamic control policies.

While carrying out our experiments for dynamic policies, we realized that there were no numerical algorithms for solving diffusion control problems in the literature that we could conveniently use for our unique setting. We thus devised two new algorithms which iterate over the average cost. The first algorithm uses binary search and thus has a linear convergence rate. The second algorithm uses the Newton-Raphson method for finding roots, and thus has a superior quadratic convergence rate. We believe that these algorithms are by themselves very useful.

## References

- [1] Kranthi Mitra Adusumilli and John J. Hasenbein. Dynamic admission and service rate control of a queue. *Queueing Syst.*, 66(2):131–154, 2010.
- [2] Rakesh Agrawal, Michael J. Carey, and Miron Livny. Models for studying concurrency control performance: alternatives and implications. *SIGMOD Rec.*, 14(4):108–121, 1985.
- [3] Baris Ata and Shiri Shneorson. Dynamic control of an  $M/M/1$  service system with adjustable arrival and service rates. *Management Science*, 52(11):1778–1791, 2006.
- [4] B. Avi-Itzhak and S. Halfin. Expected response times in a non-symmetric time sharing queue with a limited number of service positions. In *Proceedings of ITC*, 12:5.4B.2.1–7, 1988.
- [5] Robert J. Batt and Christian Terwiesch. Doctors under load: An empirical study of state-dependent service times. 2012.
- [6] D. Bertsekas. *Dynamic Programming and Optimal Control*, volume 1-2. Athena Scientific, 3rd edition, 2007.
- [7] Russ Blake. Optimal control of thrashing. In *Proceedings of the 1982 ACM SIGMETRICS Conference on Measurements and Modeling of Computer Systems*, 1982.
- [8] Maury Bramson. State space collapse with application to heavy traffic limits for multiclass queueing networks. *Queueing Syst.*, 30(1-2):89–148, 1998.
- [9] Shelby L. Brumelle. Some inequalities for parallel-server queues. *Operations Research*, 19(2):402–413, 1971.

- [10] Amarjit Budhiraja and Arka Prasanna Ghosh. Diffusion approximations for controlled stochastic networks: an asymptotic bound for the value function. *Ann. Appl. Probab.*, 16(4):1962–2006, 2006.
- [11] D.J. Daley and T. Rolski. Some comparibility results for waiting times in single- and many-server queues. *J. Appl. Prob.*, 21:887–900, 1984.
- [12] Peter J. Denning, Kevin C. Kahn, Jacques Leroudier, Dominique Potier, and Rajan Suri. Optimal multiprogramming. *Acta Informatica*, 7:197–216, 1976.
- [13] Sameh Elnikety, Erich Nahum, John Tracy, and Willy Zwaenepoel. A method for transparent admission control and request scheduling in e-commerce web sites. In *World-Wide-Web Conference*, 2004.
- [14] David Gamarnik and Petar Momčilović. Steady-state analysis of a multiserver queue in the halfin-whitt regime. *Adv. Appl. Probab.*, 40(2):548–577, 2008.
- [15] David Gamarnik and Assaf Zeevi. Validity of heavy traffic steady-state approximation in generalized Jackson networks. *Ann. Appl. Probab.*, 16(1):56–90, 2006.
- [16] Jennifer M. George and J. Michael Harrison. Dynamic control of a queue with adjustable service rate. *Operations Research*, 49(5):pp. 720–731, 2001.
- [17] Varun Gupta and Mor Harchol-Balter. Self-adaptive admission control policies for resource-sharing systems. In *Proceedings of ACM SIGMETRICS '09*, pages 311–322, New York, NY, USA, 2009.
- [18] Shlomo Halfin and Ward Whitt. Heavy-traffic limits for queues with many exponential servers. *Operations research*, 29(3):567–588, 1981.
- [19] J. Michael Harrison, Thomas M. Sellke, and Allison J. Taylor. Impulse control of Brownian motion. *Math. Oper. Res.*, 8(3):454–466, 1983.
- [20] J. Michael Harrison and Michael I. Taksar. Instantaneous control of Brownian motion. *Math. Oper. Res.*, 8(3):439–453, 1983.
- [21] Hans-Ulrich Heiss and Roger Wagner. Adaptive load control in transaction processing systems. In *Proceedings of the 17th International Conference on Large Data Bases (VLDB)*, 1991.
- [22] Onésimo Hernández-Lerma and Jean Bernard Lasserre. *Discrete-Time Markov Control Processes*. Springer, 1995.
- [23] A.J.E.M. Janssen, J.S.H. van Leeuwen, and Jaron Sanders. Scaled control in the ged regime. *Performance Evaluation*, 70(10):750–769, 2013.
- [24] Samuel Karlin and Howard M Taylor. *A Second Course in Stochastic Processes*. Academic Press, 1981.
- [25] Julian Köllerström. Heavy traffic theory for queues with several servers. I. *J. Appl. Prob.*, 11:544–552, 1974.
- [26] E. V. Krichagina and A. A. Puhalskii. A heavy-traffic analysis of a closed queueing system with a  $GI/\infty$  service center. *Queueing Syst.*, 25(1-4):235–280, 1997.

- [27] E.V. Krichagina. Asymptotic analysis of queueing networks. *Stochastics and Stochastic Reports*, 40:43–76, 1992.
- [28] Harold J. Kushner and Paul Dupuis. *Numerical Methods for Stochastic Control Problems in Continuous Time*. Springer, 2001.
- [29] Chihoon Lee and Anatolii A. Puhalskii. Non-markovian state dependent networks in critical loading. arXiv preprint arXiv:1212:4078, 2012.
- [30] Chihoon Lee and Ananda Weerasinghe. Convergence of a queueing system in heavy traffic with general patience-time distributions. *Stochastic Processes and their Applications*, 121(11):2507–2552, 2011.
- [31] Nelson Lee and VidyadharG. Kulkarni. Optimal arrival rate and service rate control of multi-server queues. *Queueing Syst.*, 76(1):37–50, 2014.
- [32] Avi Mandelbaum and Gennady Pats. State-dependent stochastic networks. part i. approximations and applications with continuous diffusion limits. *Ann. Appl. Probab.*, 8(2):569–646, 05 1998.
- [33] Petr Mandl. *Analytical Treatment of One-Dimensional Markov Processes*. Academia, Springer, 1968.
- [34] Jayakrishnan Nair, Adam Wierman, and Bert Zwart. Tail-robust scheduling via limited processor sharing. *Perform. Eval.*, 67(11):978–995, 2010.
- [35] Martin L. Puterman and Shelby L. Brumelle. On the convergence of policy iteration in stationary dynamic programming. *Mathematics of Operations Research*, 4(1):pp. 60–69, 1979.
- [36] M.L. Puterman. Optimal control of diffusion processes with reflection. *Journal of Optimization Theory and Applications*, 22(1):103–116, 1977.
- [37] Josh E Reed. The  $G/GI/N$  queue in the Halfin-Whitt regime. *Ann. Appl. Probab.*, 19(6):2211–2269, 2009.
- [38] K.M. Rege and M. Sengupta. Sojourn time distribution in a multiprogrammed computer system. *AT&T Tech. J.*, 64:1077–1090, 1985.
- [39] Amy R. Ward and Sunil Kumar. Asymptotically optimal admission control of a queue with impatient customers. *Mathematics of Operations Research*, 33(1):pp. 167–202, 2008.
- [40] Matt Welsh, David Culler, and Eric Brewer. Seda: an architecture for well-conditioned, scalable internet services. *SIGOPS Oper. Syst. Rev.*, 35(5):230–243, 2001.
- [41] Keigo Yamada. Diffusion approximation for open state-dependent queueing networks in the heavy traffic situation. *Ann. Appl. Probab.*, 5(4):958–982, 1995.
- [42] Genji Yamazaki and Hirotaka Sakasegawa. An optimal design problem for limited processor sharing systems. *Management Science*, 33(8):pp. 1010–1019, 1987.
- [43] J. Zhang and B. Zwart. Steady state approximations of limited processor sharing queues in heavy traffic. *Queueing Syst.*, 60:227–246, 2008.



- [44] Jiheng Zhang, J. G. Dai, and Bert Zwart. Law of Large Number Limits of Limited Processor-Sharing Queues. *Math. Oper. Res.*, 34(4):937–970, 2009.
- [45] Jiheng Zhang, J. G. Dai, and Bert Zwart. Diffusion Limits of Limited Processor-Sharing Queues. *Ann. Appl. Probab.*, 21(2):745–799, 2011.

## A Diffusion and Steady State Analysis for the Workload Processes

Following from the dynamic equation (7), the diffusion-scaled workload is

$$\hat{W}^{(r)}(t) = \hat{W}^{(r)}(0) + \frac{1}{r} \sum_{i=1}^{\Lambda^{(r)}(r^2 t)} v_i^{(r)} - \frac{1}{r} \int_0^{r^2 t} \mu^{(r)}(Z^{(r)}(s)) 1_{\{W^{(r)}(s) > 0\}} ds \quad (61)$$

Now, introduce the notations

$$\bar{K}^{(r)}(t, x) = \frac{1}{r^2} \sum_{i=1}^{\lceil r^2 t \rceil} 1_{\{v_i^{(r)} \leq x\}}, \quad (62)$$

$$\hat{K}^{(r)}(t, x) = r[\bar{K}^{(r)}(t, x) - tG(x)]. \quad (63)$$

The second term on the right-hand side of (61) can be written as

$$\begin{aligned} & r \int_0^t \int_0^\infty x d\bar{K}^{(r)}\left(\frac{1}{r^2} \Lambda^{(r)}(r^2 s), x\right) \\ &= \int_0^t \int_0^\infty x d\hat{K}^{(r)}\left(\frac{1}{r^2} \Lambda^{(r)}(r^2 s), x\right) + \frac{1}{r} \int_0^t \int_0^\infty x dG(x) d\Lambda^{(r)}(r^2 s) \\ &= \int_0^t \int_0^\infty x d\hat{K}^{(r)}\left(\frac{1}{r^2} \Lambda^{(r)}(r^2 s), x\right) + m \int_0^t d\frac{1}{r} [\Lambda^{(r)}(r^2 s) - \lambda r^2 s] + \lambda r m t. \end{aligned}$$

The last term on the right-hand side of (61) can be written as

$$\begin{aligned} & -r \int_0^t \mu^{(r)}(r\hat{Z}^{(r)}(s)) 1_{\{W^{(r)}(r^2 s) > 0\}} ds \\ &= \int_0^t r[\lambda m - \mu^{(r)}(r\hat{Z}^{(r)}(s))] ds + r \int_0^t 1_{\{\hat{W}^{(r)}(s)=0\}} ds - \lambda r m t \\ &= \int_0^t r[\lambda m - \mu^{(r)}(r\Delta(\hat{W}^{(r)}(s)) \wedge k^{(r)})] ds + \int_0^t r[\mu^{(r)}(r\hat{Z}^{(r)}(s)) - \mu^{(r)}(r\Delta(\hat{W}^{(r)}(s)) \wedge k^{(r)})] ds \\ &\quad + r \int_0^t 1_{\{\hat{W}^{(r)}(s)=0\}} ds - \lambda r m t \\ &= \int_0^t \theta^{(r)} \left( \Delta(\hat{W}^{(r)}(s)) \wedge \frac{k^{(r)}}{r} \right) - \theta \left( \Delta(\hat{W}^{(r)}(s)) \wedge \frac{k^{(r)}}{r} \right) ds + r \int_0^t 1_{\{\hat{W}^{(r)}(s)=0\}} ds - \lambda r m t \\ &\quad + \int_0^t r[\mu^{(r)}(r\hat{Z}^{(r)}(s)) - \mu^{(r)}(r\Delta(\hat{W}^{(r)}(s)) \wedge k^{(r)})] ds + \int_0^t \theta \left( \Delta(\hat{W}^{(r)}(s)) \wedge \frac{k^{(r)}}{r} \right) ds \end{aligned}$$

In summary, we can write the workload process as

$$\begin{aligned} \hat{W}^{(r)}(t) &= \hat{W}^{(r)}(0) + \hat{M}_s^{(r)}(t) + \hat{M}_a^{(r)}(t) + \hat{G}_1^{(r)}(t) + \hat{G}_2^{(r)}(t) \\ &\quad + \int_0^t \theta \left( \Delta(\hat{W}^{(r)}(s)) \wedge \frac{k^{(r)}}{r} \right) ds + r \int_0^t 1_{\{\hat{W}^{(r)}(s)=0\}} ds, \end{aligned} \quad (64)$$

where

$$\hat{M}_s^{(r)}(t) = \int_0^t \int_0^\infty x d\hat{K}^{(r)}\left(\frac{1}{r^2}\Lambda^{(r)}(r^2s), x\right), \quad (65)$$

$$\hat{M}_a^{(r)}(t) = m \int_0^t d\frac{1}{r}[\Lambda^{(r)}(r^2s) - \lambda r^2s], \quad (66)$$

$$\hat{G}_1^{(r)}(t) = \int_0^t r[\mu^{(r)}(r\hat{Z}^{(r)}(s)) - \mu^{(r)}(r\Delta(\hat{W}^{(r)}(s)) \wedge k^{(r)})]ds, \quad (67)$$

$$\hat{G}_2^{(r)}(t) = \int_0^t \theta^{(r)} \left( \Delta(\hat{W}^{(r)}(s) \wedge \frac{k^{(r)}}{r}) \right) - \theta \left( \Delta(\hat{W}^{(r)}(s)) \wedge \frac{k^{(r)}}{r} \right) ds. \quad (68)$$

The following lemma is an extension of the classical one-dimensional Skorohod problem. The proof can be found in [30].

**Lemma 2** *Suppose  $g$  is a Lipschitz continuous function. For any  $x \in \mathbf{D}(\mathbb{R}^+)$ , there exists a unique pair  $(y, z) \in \mathbf{D}^2(\mathbb{R}^+)$  satisfying*

$$z(t) = \int_0^t g(z(s))ds + x(t) + y(t), \quad (69)$$

$$z(t) \geq 0, \quad \text{for all } t \geq 0, \quad (70)$$

$$y(0) = 0 \text{ and } y \text{ is non-decreasing}, \quad (71)$$

$$\int_0^t z(s)dy(s) = 0. \quad (72)$$

More over, denote  $z = \psi(x)$ . The mapping  $\psi : \mathbf{D}(\mathbb{R}^+) \rightarrow \mathbf{D}(\mathbb{R}^+)$  is continuous in the uniform topology on compact set.

**Proof of Theorem 1:** We first study the first four terms on the right-hand side of equation (64). For the initial condition  $\hat{W}^{(r)}(0)$ , its convergence to some random variable  $w_0$  is part of the assumption (20) on the initial state.

According to Lemma 3.8 in [26],

$$\int_0^t \int_0^\infty x d\hat{K}^{(r)}\left(\frac{1}{r^2}\Lambda^{(r)}(r^2s), x\right) \Rightarrow \sqrt{\lambda}mc_s M_s(t), \quad \text{as } r \rightarrow \infty,$$

where  $M_s(t)$  is a standard Brownian motion (with zero drift and variance 1).

It follows from the assumption (14) that

$$\hat{M}_a^{(r)}(t) = m\hat{\Lambda}^{(r)}(t) \Rightarrow \sqrt{\lambda}mc_a M_a(t), \quad \text{as } r \rightarrow \infty.$$

We now study the terms  $\hat{G}_1^{(r)}$  and  $\hat{G}_2^{(r)}$ . By the stochastic bound (Lemma 3) proved in Section B, for any  $\epsilon > 0$ , there exists  $C$  such that  $\mathbb{P}(\Omega_r) \geq 1 - \epsilon$ , where  $\Omega_r = \left\{ \sup_{t \in [0, T]} \max \left( \hat{Z}^{(r)}(s), \Delta \hat{W}^{(r)}(s) \right) \leq C \right\}$  (noting that we naturally have  $\hat{Z}^{(r)}(\cdot) \leq k^{(r)}/r$ ). According to condition (16), for any sample path in the event  $\Omega_r$ , we have

$$\hat{G}_1^{(r)}(t) \Rightarrow 0, \quad \hat{G}_2^{(r)}(t) \Rightarrow 0, \quad \text{as } r \rightarrow \infty.$$

Let  $\hat{Y}^{(r)}(t) = r \int_0^t 1_{\{\hat{W}^{(r)}(s)=0\}} ds$ . It is easy to see that

$$\int_0^t \hat{W}^{(r)}(s) d\hat{Y}^{(r)}(s) = 0. \quad (73)$$

Thus  $(\hat{W}^{(r)}, \hat{Y}^{(r)})$  is the solution to the reflection mapping in Lemma 2. So

$$\hat{W}^{(r)} = \psi \left( \hat{W}^{(r)}(0) + \hat{M}_s^{(r)} + \hat{M}_a^{(r)} + \hat{G}_1^{(r)} + \hat{G}_2^{(r)} \right).$$

By the continuous mapping theorem,  $\hat{W}^{(r)} \Rightarrow W^*$ , where  $W^* = \psi(w_0 + \sqrt{\lambda} m c_s M_s(t) + \sqrt{\lambda} m c_a M_a(t))$ . In other words, the limit  $W^*$  satisfies

$$W^*(t) = w_0 + \sqrt{\lambda} m c_s M_s(t) + \sqrt{\lambda} m c_a M_a(t) - \theta(\Delta(W^*))(t) + Y^*(t), \quad (74)$$

with  $Y^*(0) = 0$  and being non-decreasing and

$$\int_0^t W^*(s) dY^*(s) = 0. \quad (75)$$

Thus, we have shown that the diffusion limit of the workload process is an RBM with state-dependent drift  $-\theta(\Delta_K(W^*(t)) \wedge K)$  and variance  $\lambda m^2(c_s^2 + c_a^2)$ . The proof of (25) follows immediately from the continuous mapping theorem.  $\blacksquare$

**Proof of Lemma 1:** [24, Chapter 15] prescribed an approach based on the Kolmogorov equation to compute the stationary distribution for general diffusion processes. Here we provide an alternate derivation using the basic adjoint relationship for an RBM with state-dependent drift and variance.

We can write  $W(t)$  as

$$W(t) = W(0) - \int_0^t \beta(W(\tau)) ds + \int_0^t \sqrt{s(W(\tau))} dB(\tau) + Y(t), \quad (76)$$

where  $B$  is a standard Brownian motion and  $Y$  is the regulator process that prevents  $W$  from becoming negative. The process  $Y$  is non-decreasing and satisfies

$$\int_0^t W(\tau) dY(\tau) = 0. \quad (77)$$

Let  $f$  be a twice differentiable function. By Ito's formula,

$$\begin{aligned} f(W(t)) - f(W(0)) &= \int_0^t \sqrt{s(W(\tau))} f'(W(\tau)) dB(\tau) \\ &\quad + \int_0^t \left[ \frac{1}{2} s(W(\tau)) f''(W(\tau)) - \beta(W(\tau)) f'(W(\tau)) \right] d\tau \\ &\quad + \int_0^t f'(W(\tau)) dY(\tau). \end{aligned}$$

Note that  $\int_0^t f'(W(\tau)) dB(\tau)$  is a martingale, and that

$$\int_0^t f'(W(\tau)) dY(\tau) = f'(0) Y(t),$$

due to regulation (77). Taking the conditional expectation with respect to the stationary distribution  $\pi$  on both sides of the above formula, we have

$$0 = \mathbb{E}_\pi \int_0^t \left[ \frac{1}{2} s(W(\tau)) f''(W(\tau)) - \beta(W(\tau)) f'(W(\tau)) \right] d\tau + f'(0) \mathbb{E}_\pi[Y(t)].$$

So

$$\int_0^\infty \left[ \frac{1}{2} s^2(W(\tau)) f''(w) - \beta(w) f'(w) \right] d\pi(w) + f'(0) \frac{\mathbb{E}_\pi[Y(t)]}{t} = 0. \quad (78)$$

This is known as the *basic adjoint relation* (BAR) in the literature. Our goal is to guess a functional form for  $\pi$  so that the integral in the above expression can be decomposed as  $f'(0)$  times a term independent of  $f$ .

Consider the following derivative:

$$\begin{aligned} & \left[ c(w) h(w) e^{\int_0^w g(u) du} \right]' \\ &= \left[ c(w) h'(w) + c(w) h(w) g(w) + c'(w) h(w) \right] e^{\int_0^w g(u) du} \\ &= \left[ c(w) h'(w) + [c(w) g(w) + c'(w)] h(w) \right] e^{\int_0^w g(u) du} \end{aligned}$$

Now substituting  $h(w) = f'(w)$  and  $c(w) = \frac{1}{2} s(w)$ , we obtain the expression inside the square brackets in the integral term of BAR if  $c'(w) + c(w) g(w) = -\mu(w)$ . Equivalently,  $g(w) = \frac{-\mu(w) - \frac{1}{2} s'(w)}{\frac{1}{2} s(w)}$ .

Therefore, letting  $d\pi(w) = \alpha \cdot e^{-\int_0^w \frac{\mu(v) + \frac{1}{2} s'(v)}{\frac{1}{2} s(v)} dv}$ ,

$$\begin{aligned} & \int_0^\infty \left[ \frac{1}{2} s(w) f''(w) - \mu(w) f'(w) \right] d\pi(w) \\ &= \int_0^\infty \left[ \frac{1}{2} s(w) f''(w) - \mu(w) f'(w) \right] \alpha \cdot e^{-\int_0^w \frac{\mu(v) + \frac{1}{2} s'(v)}{\frac{1}{2} s(v)} dv} \\ &= \alpha \int_0^\infty d \left[ \frac{1}{2} s(w) f'(w) e^{-\int_0^w \frac{\mu(v) + 0.5 s'(v)}{0.5 s(v)} dv} \right] \\ &= \alpha \left[ \frac{1}{2} s(\infty) f'(\infty) e^{-\int_0^\infty \frac{\mu(v) + 0.5 s'(v)}{0.5 s(v)} dv} - \frac{1}{2} s(0) f'(0) \right] \\ &= -\frac{1}{2} s(0) \alpha f'(0). \end{aligned}$$

If we plug in  $f(w) = w$  into the BAR (78), we will get  $\frac{\mathbb{E}_\pi[Y(t)]}{t} = \frac{1}{2} s(0) \alpha$ . This proves the lemma. ■

**Proof of Proposition 2:** We start with Lemma 1 and substitute state-dependent variance and drift as

$$\begin{aligned} s(w) &= \lambda m^2 (c_a^2 + c_s^2) \\ \beta(w) &= \begin{cases} \theta(w/m_e) = -\lambda m \frac{d \log f(x)}{dx} \Big|_{x=\frac{w}{m_e}} & w \leq K \cdot m_e \\ \theta(K) = -\lambda m \frac{d \log f(x)}{dx} \Big|_{x=K} & w > K \cdot m_e \end{cases} \end{aligned}$$

To obtain a further simplification, we use our assumption that  $\frac{d \log f(x)}{dx}$  is a constant for  $x \geq K$ , and therefore

$$\beta(w) = -\lambda m \frac{d \log f(x)}{dx} \Big|_{x=\frac{w}{m_e}}, \quad \forall w \in [0, \infty)$$

We then get

$$\begin{aligned}
\Pr[W^*(\infty) \leq w] &= \frac{\alpha'}{\lambda m^2(c_a^2 + c_s^2)} \int_0^w e^{\frac{2}{\lambda m^2(c_a^2 + c_s^2)} \int_0^u \lambda m d \log f(v/m_e)} du \\
&= \frac{\alpha'}{\lambda m^2(c_a^2 + c_s^2)} \int_0^w e^{\frac{2\lambda m m_e}{\lambda m^2(c_a^2 + c_s^2)} \int_0^{u/m_e} d \log f(z)} du \\
&= \frac{\alpha''}{\lambda m^2(c_a^2 + c_s^2)} \int_0^w e^{\frac{1+c_s^2}{c_a^2+c_s^2} \log f(u/m_e)} du \\
&= \frac{\alpha''}{\lambda m^2(c_a^2 + c_s^2)} \int_0^w f\left(\frac{u}{m_e}\right)^{\frac{1+c_s^2}{c_a^2+c_s^2}} du \\
&= \alpha \int_0^{\frac{w}{m_e}} f(u)^{\frac{1+c_s^2}{c_a^2+c_s^2}} du
\end{aligned}$$

which proves (29).

From (25) and the continuous mapping theorem

$$X^*(\infty) = \frac{W^*(\infty) \wedge K m_e}{m_e} + \frac{(W^*(\infty) - K m_e)^+}{m}. \quad (79)$$

It now follows that

$$\Pr[X^*(\infty) \leq x] = \begin{cases} \Pr[W^*(\infty) \leq x m_e] & x \leq K \\ \Pr[W^*(\infty) \leq K m_e + (x - K) m] & x > K \end{cases}$$

which, together with (29), gives (30).

To find  $\mathbf{E}[X^*(\infty)]$ , we will find it convenient to start with (29) and rewrite it as

$$\Pr\left[\frac{W^*(\infty)}{m_e} \leq z\right] = \alpha \int_0^z f(x)^{\frac{c_s^2+1}{c_s^2+c_a^2}} dx. \quad (80)$$

Therefore,  $f(x)^{\frac{c_s^2+1}{c_s^2+c_a^2}}$  is the density of  $\frac{W^*(\infty)}{m_e}$ . Now we again use the map (79) to write

$$\begin{aligned}
\mathbf{E}[X^*(\infty)] &= \mathbf{E}\left[\frac{W^*(\infty)}{m_e} \wedge K\right] + \frac{m_e}{m} \mathbf{E}\left[\left(\frac{W^*(\infty)}{m_e} - K\right)^+\right] \\
&= \frac{\int_0^\infty (x \wedge K) f(x)^{\frac{c_s^2+1}{c_s^2+c_a^2}} dx}{\int_0^\infty f(x)^{\frac{c_s^2+1}{c_s^2+c_a^2}} dx} + \frac{c_s^2+1}{2} \frac{\int_0^\infty (x - K)^+ f(x)^{\frac{c_s^2+1}{c_s^2+c_a^2}} dx}{\int_0^\infty f(x)^{\frac{c_s^2+1}{c_s^2+c_a^2}} dx},
\end{aligned}$$

which proves (31). ■

**Proof of Theorem 2:** This theorem essentially establishes the interchange of the steady state and heavy traffic limits for the constructed sequence of Sd-LPS models. Proving such an interchange usually involves quite a complicated analysis of a well-constructed Lyapunov function (see, for example, [15] and [30]). Taking advantage of the existing studies, we use a coupling argument to prove the interchange for our model. The proofs for both the workload and queue length essentially follow the same argument. We only focus on the queue length in this proof.

For each  $r$ , we construct an auxiliary system which takes exactly the same arrival stream as the  $r$ th Sd-LPS system and the same initial condition. Denote

$$\mu_{\dagger}^{(r)} = \mu^{(r)}(k^{(r)}).$$

When the number of jobs in the auxiliary system is more than  $k^{(r)}$ , the server works at rate  $\mu_{\dagger}^{(r)}$ . When the number of jobs drops below  $k^{(r)}$ , the server works at speed 0 (in other words it completely shuts down). Without loss of generality, we assume that the initial number of jobs is larger than  $k^{(r)}$ . Let  $Q^{(r)}(t)$  and  $Q_{\dagger}^{(r)}(t)$  denote the number of jobs in the queue in the Sd-LPS and auxiliary systems, respectively. It is clear that

$$Q^{(r)}(t) < Q_{\dagger}^{(r)}(t). \quad (81)$$

Due to parallel processing, overtaking can happen in each system, i.e., the  $j$ th arriving job may leave the system earlier than the  $i$ th arriving job even if  $j > i$ . However, due to the coupling, the  $i$ th arriving job in the auxiliary system can never enter service earlier than the corresponding job in the Sd-LPS system.

By condition (17),  $\mu_{\dagger}^{(r)} > \lambda m$  for all large enough  $r$ . So both  $Q^{(r)}$  and  $Q_{\dagger}^{(r)}$  are stationary. Let  $\pi^{(r)}$  denote the stationary probability measure of the diffusion-scaled process  $\hat{Q}^{(r)}$ . Similarly, Let  $\pi_{\dagger}^{(r)}$  denote the stationary probability measure of the diffusion-scaled queue length  $\hat{Q}_{\dagger}^{(r)}$  in the coupled system. The key step to showing that  $X^{(r)}(\infty) \Rightarrow X^*(\infty)$  as  $r \rightarrow \infty$  is to show that the family of probability measures  $\{\pi^{(r)}\}_{r \in \mathbb{N}}$  is tight. (Since  $\hat{X}^{(r)}(t) \leq \hat{Q}^{(r)}(t) + k^{(r)}/r$ , studying only the queue length suffices.) Readers can refer to the proof of Theorem 8 in [15] for a standard argument of how to prove the convergence using tightness. We now focus on proving the tightness of probability measures  $\{\pi^{(r)}\}_{r \in \mathbb{N}}$ .

We can model the  $r$ th auxiliary system as if it has  $k^{(r)}$  identical servers. All the servers either work or stop in perfect synchronization. Denote by  $S_{\dagger,i}^{(r)}(\cdot)$ ,  $i = 1, \dots, k^{(r)}$ , independent renewal processes with inter-renewal time following distribution  $G(\cdot/k^{(r)})$ , where  $G$  is the distribution of job sizes. In other words, the inter-renewal time has mean  $mk^{(r)}$  and SCV  $c_s^2$ . The queueing dynamics of the  $r$ th auxiliary system can be written as

$$Q_{\dagger}^{(r)}(t) = Q_{\dagger}^{(r)}(0) + \Lambda^{(r)}(t) - \sum_{i=1}^{k^{(r)}} S_{\dagger,i}^{(r)}(B_{\dagger}^{(r)}(t)),$$

where  $B_{\dagger}^{(r)}(t)$  is the cumulative busy time for each of the servers. Applying the diffusion scaling, we have

$$\hat{Q}_{\dagger}^{(r)}(t) = \hat{Q}_{\dagger}^{(r)}(0) + \hat{\Lambda}^{(r)}(t) - \sum_{i=1}^{k^{(r)}} \hat{S}_{\dagger,i}^{(r)}\left(\frac{1}{r^2} B_{\dagger}^{(r)}(r^2 t)\right) + r(\lambda - \mu_{\dagger}^{(r)}/m)t + \frac{\mu_{\dagger}^{(r)}}{rm}(r^2 t - B^{(r)}(r^2 t)) \quad (82)$$

where

$$\hat{\Lambda}^{(r)}(t) = \frac{1}{r} \left( \Lambda^{(r)}(r^2 t) - \lambda r^2 t \right), \quad \hat{S}_{\dagger,i}^{(r)} = \frac{1}{r} \left( S_{\dagger,i}^{(r)}(r^2 t) - \frac{\mu_{\dagger}^{(r)} r^2}{mk^{(r)}} t \right).$$

Note that  $r^2t - B^{(r)}(r^2t)$  increases only when  $\hat{Q}_{\dagger}^{(r)}(t) = 0$ , so (82) is the same as the Skorohod mapping for the  $G/G/1$  queue except that the service process is the superposition of  $k^{(r)}$  renewal processes with a much lower speed (roughly  $1/k^{(r)} \approx 1/r$ ) rather than a single renewal process. We now take advantage of the tools developed in [30] by verifying that the processes  $\hat{\Lambda}^{(r)}(t)$  and  $\hat{S}_{\dagger,i}^{(r)}$  satisfy condition (A8.p) there. That is, we want to show

$$\mathbb{E} \left[ \sup_{0 \leq s \leq t} |\hat{\Lambda}^{(r)}(s)|^2 \right] < C(1+t), \quad (83)$$

$$\mathbb{E} \left[ \sup_{0 \leq s \leq t} |\hat{S}_{\dagger,i}^{(r)}(s)|^2 \right] < \frac{C}{r}(1+t). \quad (84)$$

Condition (83) directly follows from assumption (14), following the same argument as in [30] (essentially using Lemma 3.5 in [10]). To verify (84), we need to further investigate the proof of Lemma 3.5 in [10]. It follows from (3.31) and (3.32) in the proof that the result of Lemma 3.5 can be enhanced as follows: The right-hand side of the second inequality in (3.20), which is  $C^*(1+t)$ , can be replaced by  $\frac{2}{r} + \frac{C_2}{r^2} + 3C_2t$ . Since our renewal process  $S_{\dagger,i}^{(r)}$  has speed  $1/k^{(r)}$  rather than 1, by time change, we can replace the time  $t$  by  $\frac{t}{k^{(r)}}$ . So  $\mathbb{E} \left[ \sup_{0 \leq s \leq t} |\hat{S}_{\dagger,i}^{(r)}(s)|^2 \right] < \frac{2}{r} + \frac{C_2}{r^2} + 3C_2 \frac{t}{k^{(r)}} \leq \frac{3}{r} + 6KC_2 \frac{t}{r}$  for all large enough  $r$ . This proves (84). Then following exactly the same argument, Theorem 3.3 in [30] holds for our problem. This implies Theorem 3.2 in [30], i.e.,  $\sup_r \int_0^\infty w \pi_{\dagger}^{(r)}(dw) < \infty$ . By the coupling construction (81), we have  $\int_0^\infty x \pi^{(r)}(dx) < \int_0^\infty x \pi_{\dagger}^{(r)}(dx)$ . This implies tightness of  $\{\pi^{(r)}\}_{r \in \mathbb{N}}$ . ■

## B State Space Collapse for the Sd-LPS system

We introduce a strengthened version of the mapping  $\Delta_K$  as the follows. Let  $\Delta_{K,\nu} : \mathbb{R}_+ \rightarrow \mathbf{M} \times \mathbf{M}$  be the lifting map associated with the probability measure  $\nu$  and constant  $K$  given by

$$\Delta_{K,\nu} w = \left( \frac{(w - K\beta_e)^+}{\beta} \nu, \frac{w \wedge K\beta_e}{\beta_e} \nu_e \right) \quad \text{for } w \in \mathbb{R}_+.$$

We aim to prove the following full version of the SSC

**Theorem 3 (Full State Space Collapse)** *Under the conditions (14)–(16) and (19)–(21), for any  $T > 0$ ,*

$$\sup_{t \in [0, T]} \mathbf{d}[(\hat{Q}^{(r)}(t), \hat{Z}^{(r)}(t)), \Delta_{K,\nu} \hat{W}^{(r)}(t)] \Rightarrow 0 \quad \text{as } r \rightarrow \infty.$$

It is clear that Theorem 3 implies Proposition 1. The rest of this section is devoted to the proof of the full SSC.

### B.1 Tightness of Shifted Fluid-Scaled Processes

The key to proving SSC, which was originally developed by [8], is to “chop” the diffusion-scaled processes into pieces.

**Shifted Fluid Scaling** Introduce,

$$\bar{Q}^{(r,l)}(t) = \frac{1}{r} Q^{(r)}(rl + rt), \quad \bar{Z}^{(r,l)}(t) = \frac{1}{r} Z^{(r)}(rl + rt), \quad (85)$$



for all  $m \in \mathbb{N}$  and  $t \geq 0$ . To see the relationship between these two scalings, consider the diffusion-scaled process on the interval  $[0, T]$ , which corresponds to the interval  $[0, r^2 T]$  for the unscaled process. Fix a constant  $L > 1$ , the interval will be covered by  $\lfloor rT \rfloor + 1$  overlapping intervals

$$[rl, rl + rL] \quad l = 0, 1, \dots, \lfloor rT \rfloor.$$

For each  $t \in [0, T]$ , there exists an  $l \in \{0, \dots, \lfloor rT \rfloor\}$  and  $s \in [0, L]$  (which may not be unique) such that  $r^2 t = rl + rs$ . Thus

$$\hat{Q}^{(r)}(t) = \bar{Q}^{(r,l)}(s), \quad \hat{Z}^{(r)}(t) = \bar{Z}^{(r,l)}(s). \quad (86)$$

This will serve as a key relationship between fluid and diffusion-scaled processes.

The quantities  $Q^{(r)}(\cdot)$ ,  $Z^{(r)}(\cdot)$ ,  $X^{(r)}(\cdot)$ ,  $W^{(r)}(\cdot)$  are essentially functions of  $(\bar{Q}^{(r)}(\cdot), \bar{Z}^{(r)}(\cdot))$ , so the scaling for these quantities is defined as the functions of the corresponding scaling for  $(\bar{Q}^{(r)}(\cdot), \bar{Z}^{(r)}(\cdot))$ . For example

$$\bar{W}^{(r,l)}(t) = \langle \chi, \bar{Q}^{(r,l)}(t) + \bar{Z}^{(r,l)}(t) \rangle = \frac{1}{r} W^{(r)}(rl + rt).$$

We define the shifted fluid scaling for the arrival process as

$$\bar{\Lambda}^{(r,l)}(t) = \frac{1}{r} \Lambda^{(r)}(rl + rt),$$

for all  $t \geq 0$ . By (6), the shifted fluid scaling for  $B^{(r)}(\cdot)$  is

$$\bar{B}^{(r,l)}(t) = \bar{E}^{(r,l)}(t) - \bar{Q}^{(r,l)}(t),$$

for all  $t \geq 0$ . A shifted fluid-scaled version of the stochastic dynamic equations (4) and (5) can be written as, for any  $A \subset (0, \infty)$ ,  $0 \leq s \leq t$ ,

$$\begin{aligned} \bar{Q}^{(r,l)}(t)(A) &= \bar{Q}^{(r,l)}(s)(A) + \frac{1}{r} \sum_{i=r\bar{E}^{(r,l)}(s)+1}^{r\bar{E}^{(r,l)}(t)} \delta_{v_i}(A) \\ &\quad - \frac{1}{r} \sum_{i=r\bar{B}^{(r,l)}(s)+1}^{r\bar{B}^{(r,l)}(t)} \delta_{v_i}(A), \end{aligned} \quad (87)$$

$$\begin{aligned} \bar{Z}^{(r,l)}(t)(A) &= \bar{Z}^{(r,l)}(s)(A + S^{(r)}(rl + rs, rl + rt)) \\ &\quad + \frac{1}{r} \sum_{i=r\bar{B}^{(r,l)}(s)+1}^{r\bar{B}^{(r,l)}(t)} \delta_{v_i^{(r)}}(A + S^{(r)}(\tau_i^{(r)}, rl + rt)). \end{aligned} \quad (88)$$

We point out that the cumulative service process  $S^{(r)}$  is never scaled because it tracks the amount of service received by each individual customer. However, via some algebra we can see that

$$S^{(r)}(rl + rs, rl + rt) = \int_{rl+rs}^{rl+rt} \frac{\mu^{(r)}(Z^{(r)}(\tau))}{Z^{(r)}(\tau)} d\tau = \int_s^t \frac{\mu^{(r)}(r\bar{Z}^{(r,l)}(\tau))}{\bar{Z}^{(r,l)}(\tau)} d\tau. \quad (89)$$

This gives two interesting observations. First, the shifted fluid scaling is essentially fluid scaling, meaning the shifted fluid-scaled processes should be close to some fluid model solutions. Second, the corresponding fluid model is essentially the same as the fluid model in [45] since by (16),

$$\mu^{(r)}(r\bar{Z}^{(r,l)}(\tau)) = 1 + O^+\left(\frac{1}{r}\right),$$

where  $O^+(1/r)$  means the quantity is positive and of the same order as  $1/r$  when  $r \rightarrow \infty$ . So

$$S^{(r)}(rl + rs, rl + rt) = \int_s^t \frac{1}{\bar{Z}^{(r,l)}(\tau)} d\tau + O^+\left(\frac{1}{r}\right). \quad (90)$$

Intuitively,  $\bar{Z}^{(r,l)}$  is close to some fluid limit denoted by  $\tilde{Z}$  as  $r$  becomes very large (in the mathematical sense of convergence in probability), then

$$S^{(r)}(rl + rs, rl + rt) \Rightarrow \int_s^t \frac{1}{\tilde{Z}(\tau)} d\tau. \quad (91)$$

So we can conclude that the underlying fluid is the same as the one for the regular LPS system. Thus, we can use existing properties developed in [45]. We hope to make the argument rigorous and concise in the follows.

### Some Bound Estimation

The tightness property, which guarantees that the shifted fluid-scaled process  $\{\bar{Q}^{(r,l)}, \bar{Z}^{(r,l)}\}$  has a convergent subsequence, can be proved in a similar way as in [45]. There are two key differences. First is the service process as pointed out before. Second is that [45] heavily relies on the known result on the diffusion of the workload (see Proposition 2.1). However, we do not have such a diffusion limit of workload a priori. Instead, we try to prove such a diffusion limit by SSC. Looking into the details of the machinery in [45], what essentially is needed for the workload process is some kind of stochastic bound, which we prove in the following lemma.

**Lemma 3 (An Upper Bound of the Workload)** *For any  $\eta > 0$  there exists a constant  $M$  such that*

$$\mathbb{P} \left( \max_{l \leq rT} \sup_{t \in [0, L]} \bar{W}^{(r,l)}(t) < M \right) > 1 - \eta. \quad (92)$$

**Proof:** Using the relationship between the shifted fluid scaling and diffusion scaling, we essentially need to prove that

$$\mathbb{P} \left( \sup_{t \in [0, L]} \hat{W}^{(r)}(t) < M \right) > 1 - \eta.$$

Recall the representation (64) for the diffusion-scaled workload processes. Let  $\underline{\theta} = \inf_{x \in [0, K]} \theta(x)$ , which is finite due to condition (16), so the process  $\hat{W}_1^{(r)}$  satisfying

$$\hat{W}_1^{(r)}(t) = \hat{W}^{(r)}(0) - \underline{\theta}t + \hat{M}_s^{(r)}(t) + \hat{M}_a^{(r)}(t) + \hat{G}_1^{(r)}(t) + \hat{G}_2^{(r)}(t) + r \int_0^t 1_{\{\hat{W}_1^{(r)}(s)=0\}} ds$$

is an upperbound of  $\hat{W}^{(r)}$  due to the definition of  $\underline{\theta}$  and condition (17). By Lemma 2,  $\hat{W}_1^{(r)}$  converges to a driftless RBM, which is stochastically bounded. This implies the result. ■

Such a stochastic bound of the workload process helps to establish some useful bound estimates for the stochastic processes underlying the Sd-LPS model.

**Lemma 4 (Further Bound Estimations)** *For any  $\eta > 0$ , there exists a constant  $M > 0$  and a probability event  $\Omega_B^r(M)$  for each index  $r$  such that*

$$\liminf_{r \rightarrow \infty} \mathbb{P}(\Omega_B^r(M)) \geq 1 - \eta, \quad (93)$$

and on the event  $\Omega_B^r(M)$ , we have

$$\max_{l \leq \lfloor rT \rfloor} \sup_{t \in [0, L]} \bar{Q}^{(r, l)}(t) \leq M, \quad (94)$$

$$\max_{l \leq \lfloor rT \rfloor} \sup_{t \in [0, L]} \langle \chi^{1+p}, \bar{Q}^{(r, l)}(t) + \bar{Z}^{(r, l)}(t) \rangle \leq M. \quad (95)$$

**Proof:** The result (94) holds due to Lemma 4.2 in [45], which only utilizes the regularity of the arrival process (14) and the stochastic bound (92) for the workload process proved in Lemma 3. For (95), the first half,  $\max_{l \leq \lfloor rT \rfloor} \sup_{t \in [0, L]} \langle \chi^{1+p}, \bar{Q}^{(r, l)}(t) \rangle \leq M$ , also follows the same reasoning as Lemma 4.3 in [45]. Essentially, any results for the “queue” part follows the same argument in [45].

The challenge with the state-dependent service rate lies in the analysis of the server. It follows from the shifted fluid-scaled dynamic equation (88) that for any Borel set  $A \subset (0, \infty)$ ,

$$\begin{aligned} \frac{1}{r} \mathcal{Z}^{(r)}(rl + rt)(A) &= \frac{1}{r} \mathcal{Z}^{(r)}(0)(A + S^{(r)}(0, rl + rt)) \\ &\quad + \sum_{j=0}^{m-1} \frac{1}{r} \sum_{i=B^{(r)}(r(l-j-1))+1}^{B^{(r)}(r(l-j))} \delta_{v_i}(A + S^{(r)}(\tau_i^{(r)}, rl + rt)) \\ &\quad + \frac{1}{r} \sum_{i=B^{(r)}(rl)+1}^{B^{(r)}(rl+rt)} \delta_{v_i}(A + S^{(r)}(\tau_i^{(r)}, rl + rt)). \end{aligned}$$

Given  $0 \leq j \leq m-1$ , for those  $i$ 's with  $B^{(r)}(r(l-j-1)) < i \leq B^{(r)}(r(l-j))$  we have

$$\tau_i^{(r)} \in [r(l-j-1), r(l-j)].$$

For the sake of simplicity, let us assume that  $Z^{(r)}(s) > 0$  for all  $s \in [0, rl + rt]$ . If this does not hold, we can use a technical trick presented in the proof of Lemma 4.3 in [45] to deal with it. Here we show the main difference coming from the state-dependent service rate. By (90) and the fact that  $Z^{(r)} \leq k^{(r)}$ , we have a lower bound on the cumulative service amount

$$S^{(r)}(rs, rt) \geq \int_{rs}^{rt} \frac{1}{Z^{(r)}(s)} ds \geq \frac{r(t-s)}{k^{(r)}}. \quad (96)$$

Thus,

$$S^{(r)}(\tau_i^{(r)}, rl + rt) \geq S^{(r)}(r(l-j), rl) \geq \frac{rj}{k^{(r)}} \geq \frac{j}{2K},$$

for all large  $r$  where the last inequality is due to (9). For those  $i$ 's such that  $\tau_i^{(r)}$  is larger than  $B^{(r)}(rl)$ , we use the trivial lower bound  $S^{(r)}(\tau_i^{(r)}, rl + rt) \geq 0$ . Also take the trivial lower bound that  $S^{(r)}(0, rl + rt) \geq 0$ . Then we have the following inequality on the  $(1+p)$ th moment:

$$\begin{aligned} \langle \chi^{1+p}, \frac{1}{r} \mathcal{Z}^{(r)}(rl + rt) \rangle &\leq \langle \chi^{1+p}, \frac{1}{r} \mathcal{Z}^{(r)}(0) \rangle \\ &\quad + \sum_{j=0}^{m-1} \langle ((\chi - \frac{j}{2K})^+)^{1+p}, \frac{1}{r} \sum_{i=B^{(r)}(r(l-j-1))+1}^{B^{(r)}(r(l-j))} \delta_{v_i} \rangle \\ &\quad + \langle \chi^{1+p}, \frac{1}{r} \sum_{i=B^{(r)}(rl)+1}^{B^{(r)}(rl+rt)} \delta_{v_i} \rangle. \end{aligned} \quad (97)$$

This is the same as (4.22) in [45]. The estimation of the first term on the right-hand side in the above follows directly from the initial condition (20). The analysis of the second and third terms follows the same way as in [45]. ■

To prove that a family of measure-valued processes is tight, there are three properties to verify, namely *Compact Containment*, *Asymptotic Regularity* and *Oscillation Bound*. For brevity, we will not repeat the exact mathematical statements and their proofs. For the LPS system, these three properties were proved in Lemmas 4.4–4.6 in [45]. We just point out that the proof for the above mentioned three properties for the Sd-LPS system relies on (a) the bound estimate in Lemma 4; and (b) the fact that (90) implies the lower bound of the cumulative service process (96). The proof of Lemma 4 has demonstrated point (b) clearly, we therefore omit a repeat of the argument used in [45]. So we reach the conclusion:

**Proposition 7 (Tightness of Shifted Fluid-scaled Processes)** *The family of shifted fluid-scaled processes  $\{(\bar{Q}^{(r,l)}, \bar{Z}^{(r,l)})\}_{l \leq rT, r \in \mathbb{N}}$  is tight.*

Loosely speaking, tightness means that any subsequence from the family of shifted fluid-scaled processes has a convergent subsequence. This is formally stated in Theorem 4.1 in [45].

## B.2 Bramson’s Framework for SSC

Sd-LPS and LPS essentially use the same measure-valued framework. The difference lies in the cumulative service process as we explained when deriving (90) and the workload process as we studied in Lemma 3. After obtaining the tightness, we can apply the framework invented by Bramson [8] in the same way as how Section 5 in [45] applies it to the measure-valued process. The high level-logic is as the follows: the shifted fluid-scaled processes are “close” to the fluid model solution, and the fluid model solution converges to some invariant which exhibits SSC (Theorem 3.1 in [45]). Thus SSC, which happens on the diffusion scaling, can be proved based on the relationship (86) between diffusion scaling and shifted fluid scaling. We thus refer to Section 5 in [45] for the proof of Theorem 3.

## C Analysis of Algorithms for Finding Optimal Control

Recall some notation and definitions used in this section.

$$\begin{aligned}\hat{k} &= \arg \max_k \theta(k) \\ \hat{\theta} &= \theta(\hat{k}) \\ \Delta_k(w) &= \frac{w}{m} + k \left(1 - \frac{m_e}{m}\right) \\ d_\theta &= \sup_k \theta(k) - \inf_k \theta(k) \\ k_f(w) &= \arg \max_{k \in [0, w/m_e]} \theta(k_f)\end{aligned}$$

Throughout this section, we assume that  $d_\theta$  is finite, and therefore  $\theta(k)$  is bounded from above and below.

### C.1 Some Auxiliary Results

We first provide some auxiliary results (Lemmas 5 and 6) which will be useful in proving the results in Section 4.

**Lemma 5** Consider the solution of the following ODE, parameterized by  $v$  and  $W$ :

Terminal condition:

$$G_{v,W}(w) = \alpha w + \beta v + \gamma \quad \dots w \geq \max\{W, \hat{k}m_e\}$$

ODE:

$$\begin{aligned} v &= \frac{w}{m} + k_f(w) \left(1 - \frac{m_e}{m}\right) - \theta(k_f(w))G_{v,W}(w) + \frac{\sigma^2}{2}G'_{v,W}(w) & \dots w \in [W, \hat{k}m_e] \\ v &= \min_{k \in [0, w/m_e]} \left\{ \frac{w}{m} + k \left(1 - \frac{m_e}{m}\right) - \theta(k)G_{v,W}(w) + \frac{\sigma^2}{2}G'_{v,W}(w) \right\} & \dots w \in [0, W] \end{aligned}$$

Then  $G_{v,W}(w)$  is continuous in both  $v$  and  $W$  for all  $w$ .

**Proof:** Let  $(v_a, W_a)$  and  $(v_b, W_b)$  denote two parameter settings, and for succinctness, denote the corresponding solutions to the ODE as  $G_a$  and  $G_b$ , respectively. We will consider the case  $W_a, W_b \geq \hat{k}m_e$  as other cases are analogous.

Let  $W_a \leq W_b$ .

At  $w = W_b$ , we have

$$|G_a(W_b) - G_b(W_b)| = \beta|v_a - v_b|. \quad (98)$$

For  $w \in [W_a, W_b]$ , we have

$$G_a(w) = \alpha w + \beta v_a + \gamma, \quad (99)$$

$$G'_b(w) = \frac{2}{\sigma^2} \left( v_b + \theta(k_b(w))G_b(w) - \frac{w}{m} + k_b \left(1 - \frac{m_e}{m}\right) \right), \quad (100)$$

which gives

$$\frac{2}{\sigma^2} \left( v_b - \frac{W_b}{m \wedge m_e} - d_\theta G_b(w) \right) \leq G'_b(w) \leq \frac{2}{\sigma^2} \left( v_b - \frac{W_a}{m \vee m_e} + d_\theta G_b(w) \right). \quad (101)$$

Since the derivatives are bounded,  $G_b(w)$  is bounded in the interval  $[W_a, W_b]$ . Let  $D = \sup_{w \in [W_a, W_b]} |G_b(w)|$ . Then,

$$|G_a(w) - G_b(w)| \leq |G_a(w) - G_a(W_a)| + |G_a(W_b) - G_b(W_b)| + |G_b(w) - G_b(W_b)| \quad (102)$$

$$\leq \alpha|W_b - w| + \beta|v_a - v_b| + (W_b - w) \frac{2}{\sigma^2} \left( v_b + \frac{W_b}{m \wedge m_e} + d_\theta D \right) \quad (103)$$

$$\leq \alpha|W_b - W_a| + \beta|v_a - v_b| + (W_b - W_a) \frac{2}{\sigma^2} \left( v_b + \frac{W_b}{m \wedge m_e} + d_\theta D \right), \quad (104)$$

which goes to 0 as  $|v_a - v_b| + |W_a - W_b| \rightarrow 0$ .

For  $w \in [0, W_a]$ , by Lemma 7,

$$\begin{aligned} |G'_a(w) - G'_b(w)| &\leq \frac{2}{\sigma^2}|v_a - v_b| + \frac{2}{\sigma^2} \left| \min_{k_a \in [0, w/m_e]} (k_b(1 - m_e/m) - \theta(k_a)G_a(w)) \right. \\ &\quad \left. - \min_{k_b \in [0, w/m_e]} (k_b(1 - m_e/m) - \theta(k_b)G_b(w)) \right| \\ &\leq \frac{2}{\sigma^2}|v_a - v_b| + \frac{2d_\theta}{\sigma^2}|G_a(w) - G_b(w)|. \end{aligned} \quad (105)$$

Applying Gronwall's inequality, for all  $w \in [0, W_a]$

$$|G_a(w) - G_b(w)| \leq |G_a(W_a) - G_b(W_a)| e^{\frac{2d_\theta}{\sigma^2}(W_a-w)} + \frac{|v_a - v_b|}{d_\theta} \left( e^{\frac{2d_\theta}{\sigma^2}(W_a-w)} - 1 \right),$$

which, together with (104), implies that for all  $w \in [0, W_a]$

$$\begin{aligned} |G_a(w) - G_b(w)| &\leq |v_a - v_b| \left( \left| \beta \right| e^{\frac{2d_\theta}{\sigma^2}(W_a-w)} + \frac{e^{\frac{2d_\theta}{\sigma^2}(W_a-w)} - 1}{d_\theta} \right) \\ &\quad + |W_b - W_a| \left( \alpha + \frac{2}{\sigma^2} \left( v_b + \frac{W_b}{m \wedge m_e} + d_\theta D \right) \right) e^{\frac{2d_\theta}{\sigma^2}(W_a-w)}, \end{aligned}$$

which goes to 0 as  $|v_a - v_b| + |W_a - W_b| \rightarrow 0$ . ■

**Lemma 6** Consider  $G_{v,W}$  defined in Lemma 5 for a given  $W \geq \hat{k}m_e$ . Then  $G_{v,W}(w)$  is monotonic and Lipschitz continuous in  $v$  for all  $w$ .

**Proof:** Fix  $W \geq \hat{k}m_e$ , and consider  $v_a > v_b$ . Let  $G_a$  and  $G_b$  denote the solutions of the ODE defined in Lemma 5 for  $v_a$  and  $v_b$ , respectively. We will show that  $G_a(w) < G_b(w)$  for all  $w \geq 0$ . We rely on the following two facts:

1. Terminal condition:

$$G_b(w) - G_a(w) = -\beta(v_a - v_b) \quad w \geq W$$

2. Bounds on  $G'_b(w) - G'_a(w)$  for  $w \in [0, W]$ :

$$G'_b(w) - G'_a(w) = -\frac{2}{\sigma^2} \left[ (v_a - v_b) - \min_{k \in [0, w/m_e]} (\Delta_k(w) - \theta(k)G_a(w)) - \min_{k \in [0, w/m_e]} (\Delta_k(w) - \theta(k)G_b(w)) \right]$$

where recall that  $\Delta_k(w) = \frac{w}{m} + k \left( 1 - \frac{m_e}{m} \right)$ . Under the assumption  $G_a(w) \leq G_b(w)$ , from Lemma 7:

$$-\frac{2}{\sigma} [(v_a - v_b) + d_\theta(G_b(w) - G_a(w))] \leq G'_b(w) - G'_a(w) \leq -\frac{2}{\sigma^2} [(v_a - v_b) - d_\theta(G_b(w) - G_a(w))] \quad (106)$$

Combining these two facts, we get for any  $w \in [0, W]$

$$(v_a - v_b) \left[ -\beta + \frac{1}{d_\theta} \left( 1 - e^{-\frac{2d_\theta W}{\sigma^2}} \right) \right] \leq G_b(w) - G_a(w) \leq (v_a - v_b) \left[ -\beta + \frac{1}{d_\theta} \left( e^{\frac{2d_\theta W}{\sigma^2}} - 1 \right) \right] \quad (107)$$

**Lemma 7** Let  $x_1 = \arg \min_{x \in [u, v]} f_1(x)$  and  $x_2 = \arg \min_{x \in [u, v]} f_2(x)$ . Then,

$$|f_1(x_1) - f_2(x_2)| \leq \sup_{x \in [u, v]} |f_1(x) - f_2(x)|$$

**Proof:** Proof Since  $f_1(x_1) \leq f_1(x_2)$  and  $f_2(x_2) \leq f_2(x_1)$ ,

$$f_1(x_1) - f_2(x_1) \leq f_1(x_1) - f_2(x_2) \leq f_1(x_2) - f_2(x_2)$$

and therefore,  $|f_1(x_1) - f_2(x_2)| \leq \sup_{x \in [u, v]} |f_1(x) - f_2(x)|$ . ■

## C.2 Proofs of Results in Section 4

**Proof of Proposition 3:** We should point out that the monotonicity of the value function is not immediate because under the optimal policy  $k^*(\cdot)$ , the state-dependent cost function  $\Delta_{k^*}(w)$  need not be monotonic in  $w$ . If it were, a simple sample path coupling argument could be used to deduce the monotonicity of the discounted value function by considering initial workloads  $w_1 \leq w_2$ .

Let  $k_\gamma^*(\cdot)$  be the optimal policy minimizing expected discounted cost, and  $V_\gamma(w)$  be the corresponding value function. Consider  $w_1 \leq w_2$ . We will create an alternate control policy  $\pi_1$  when the initial workload is  $w_1$ , and denote the corresponding expected discounted cost by  $\tilde{v}_1$ . We will then show that  $\tilde{v}_1 \leq V_\gamma(w_2)$  (in fact, our construction involves stochastic coupling and implies that the discounted reward starting with  $w_1$  and using  $\pi_1$  is stochastically smaller than the discounted reward starting with  $w_2$  and using  $k_\gamma^*(\cdot)$ ).

Construction of  $\pi_1$ : We simulate two independent systems in parallel: system 1 with initial workload  $W_1(0) = w_1$  under control policy  $\pi_1$  (which we will describe shortly); and system 2 with initial workload  $W_2(0) = w_2$  under the optimal control policy  $k_\gamma^*(w)$ . The control at time  $t$  under  $\pi_1$  is chosen to be

$$k_{\pi_1}(t) = \arg \min_{k \in [0, W_1(t)/m_e]} \frac{W_1(t)}{m} + k(1 - m_e/m)$$

for  $t \in [0, \tau]$ , where  $\tau \doteq \min\{s \geq 0 : W_1(s) = W_2(s)\}$  is the coupling time of the two systems. That is,  $\tau$  is the first time the workloads of the two coupled processes  $W_1$  and  $W_2$  coincide. For  $t \geq \tau$ ,  $k_{\pi_1}(t) = k_\gamma^*(W_1(t))$ .

It is easy to see that since  $W_1$  and  $W_2$  have continuous sample paths,  $W_1(t) \leq W_2(t)$  for  $t \leq \tau$ . Due to the choice of  $k_{\pi_1}$ , this further implies that

$$\begin{aligned} \min_{k \in [0, W_1(t)/m_e]} \left( \frac{W_1(t)}{m} + k \left( 1 - \frac{m_e}{m} \right) \right) &= \min \left\{ \frac{W_1(t)}{m}, \frac{W_1(t)}{m_e} \right\} \\ &\leq \min \left\{ \frac{W_2(t)}{m}, \frac{W_2(t)}{m_e} \right\} \\ &\leq \frac{W_2(t)}{m} + k_\gamma^*(W_2(t)) \left( 1 - \frac{m_e}{m} \right). \end{aligned}$$

For  $t \geq \tau$ ,  $W_1(t)$  is stochastically equal to  $W_2(t)$ . Therefore, the discounted cost of  $\pi_1$  (with initial workload  $w_1$ ) is stochastically smaller than the discounted cost of  $k_\gamma^*$  (with initial workload  $w_2$ ). This implies  $\tilde{v}_1 \leq V_\gamma(w_2)$ , but  $V_\gamma(w_1) \leq \tilde{v}_1$  (since  $V_\gamma(w_1)$  is the optimal expected discounted cost). Therefore,  $V_\gamma(w_1) \leq V_\gamma(w_2)$  when  $w_1 \leq w_2$ .

Since  $(V_\gamma(w_2) - V_\gamma(w_1)) \geq 0$  for all  $\gamma$ , this also holds as  $\gamma \downarrow 0$ .

**Note :** The only facts we relied on to argue monotonicity were (i) continuity of sample paths, and (ii) the cost of the cheapest action available in each state is monotonic in  $w$ . These appear to be weaker than the conditions typically used in the literature where the set of available actions is assumed to be independent of the state. Further, the cost is assumed to be non-decreasing in the state variable for each action. ■

We now provide the proofs of Proposition 4 and 6 for the analysis of our algorithms. We omit the proof of Proposition 5 as it mirrors the proof of Proposition 4.

**Proof of Proposition 4:** Consider the diffusion control formulation for the Sd-LPS system but with a finite workload buffer of  $W$ . For the diffusion corresponding to this loss system, we have

reflections at both  $w = 0$  and  $w = W$ . Therefore, for any policy for this loss system, the value function gradient is 0 at both these values [33]:

$$G(0) = G(W) = 0.$$

Therefore, (46) defines the HJB equation for the value function gradient of the finite buffer system with workload buffer  $W$ , together with  $G_v(0) = 0$  and the additional boundary condition  $G_v(W) = 0$ . Lemma 5 guarantees that ODE (46) has a unique solution (by choosing the terminal condition  $G_{v,W}(W) = 0$ ).

We first show that for all  $v < v^*$ , there is a *unique* value of  $W$  such that  $v$  is the average cost of the optimal finite buffer policy with workload buffer  $W$ .

Consider an arbitrary pair  $W, v$  and solve the following ODE

$$v = \min_{k \in [0, w/m_e]} \left\{ \frac{w}{m} + k \left( 1 - \frac{m_e}{m} \right) - \theta(k) G_{v,W}(w) \right\} + \frac{\sigma^2}{2} G'_{v,W}(w) \quad (108)$$

backwards with terminal condition  $G_{v,W}(W) = 0$  (note that this is the same ODE as (46) but we do not enforce  $G_{v,W}(0) = 0$ ). Lemma 6 then shows that  $G_{v,W}(0)$  is monotonic in  $v$ . Therefore, for each  $W$ , there exists a unique  $v^*(W)$  such that  $G_{v^*(W),W}(0) = 0$  for the ODE above, with terminal condition  $G_{v^*(W),W}(W) = 0$ . Further, Lipschitz continuity and Lemma 5 imply that the map  $v^*(W)$  is continuous. From the foregoing discussion, we see that  $v^*(W)$  denotes the cost of the optimal finite buffer policy with finite buffer  $W$ .

We next show that  $v = v^*(W_1) \neq v^*(W_2)$  if  $W_1 \neq W_2$ . This would imply that two different workload buffer sizes must yield different optimal costs. Assume the contrapositive, and further  $W_1 < W_2$ . This implies that  $G_{v,W_1}(w) = G_{v,W_2}(w)$  for  $w \in [0, W_1]$  when the  $G_{v,W}$  ODEs are evolved forward with initial condition  $G_{v,W_1} = G_{v,W_2} = 0$ . Then by (108),

$$\begin{aligned} G'_{v,W_2}(w)|_{w=W_1} &= G'_{v,W_1}(w)|_{w=W_1} = \frac{2}{\sigma^2} \left[ v - \min_{k \in [0, W_1/m_e]} \left( \Delta_k(W_1) - \theta(k) G_{v,W_1}(W_1) \right) \right] \\ &= \frac{2}{\sigma^2} \left[ v - \min_{k \in [0, W_1/m_e]} \Delta_k(W_1) \right] \\ &= \frac{2}{\sigma^2} \left[ v - \min \left\{ \frac{W_1}{m}, \frac{W_1}{m_e} \right\} \right] < 0. \end{aligned}$$

The last inequality is true because  $\theta()$  is bounded from below, and hence for the optimal policy with buffer  $W_1$ , the average cost is strictly smaller than  $\min\{W_1/m, W_1/m_e\}$ . This implies  $G_{v,W_2}(W_1 + \epsilon) < 0$  for any  $\epsilon > 0$ . A similar argument as in Proposition 3 shows that the optimal value function for the finite buffer system is monotonic and hence  $G_{v,W_2}(w) \geq 0$ ,  $w \in [0, W_2]$ , which contradicts  $G_{v,W_2}(W_1) = 0$  and  $G'_{v,W_2}(W_1) < 0$ .

Therefore,  $\underline{W}(v)$  as defined in (47) (if it exists) is the unique buffer size corresponding to the optimal finite buffer policy with average cost  $v$ . To get a control on how  $\underline{W}(v)$  grows as  $v \uparrow v^*$  (where  $v^*$  denotes the average cost of the infinite buffer control), we will next argue that  $\underline{W}(v) = O\left(\log \frac{1}{v^* - v}\right)$  (and hence also finite). We will instead prove the following equivalent result: let  $v_W^*$  denote the average cost of the optimal finite buffer control with workload buffer limit  $W$ , then  $(v^* - v_W^*) = O(e^{-\beta W})$  as  $W \rightarrow \infty$  for some constant  $\beta > 0$ .

Intuitively, the service rate of the optimal control must asymptotically approach  $\hat{\theta}$  as the backlog builds up, and hence the distribution of the workload (and therefore number of jobs in the system)



should decay at an exponential rate. Therefore, the effect of truncation at workload  $W$  for a finite buffer system, that is  $(v^* - v_W^*)$ , should also be  $O(e^{-\beta W})$  for some constant  $\beta > 0$ .

The proof will proceed in several steps:

**Step 1:** For the optimal infinite buffer control, there exists a constant  $\alpha$  such that the optimal value function gradient  $G^*(w)$  satisfies

$$G^*(w) \leq \alpha + \frac{w}{m\hat{\theta}}. \quad (109)$$

*Proof:* Consider the upper envelope function  $\bar{G}_{v^*}(\cdot)$  given by

$$\bar{G}_{v^*}(w) = \frac{w}{m\hat{\theta}} + \left( \hat{k} \left( 1 - \frac{m_e}{m} \right) + \frac{\sigma^2}{2m\hat{\theta}} - v^* \right) \frac{1}{\hat{\theta}}, \quad w \geq \hat{k}m_e. \quad (110)$$

For  $w \in [0, \hat{k}m_e]$ ,  $\bar{G}_{v^*}$  is obtained by solving ODE (51) backwards starting with the terminal condition

$$\bar{G}_{v^*}(\hat{k}m_e) = \left( \hat{k} - v^* + \frac{\sigma^2}{2m\hat{\theta}} \right) \frac{1}{\hat{\theta}}.$$

Note that this is the same upper envelope function we use for detecting feasibility in the binary search algorithm. Now  $v^* \leq v_f(0)$  implies  $\bar{G}_{v^*}(0) \geq 0$  since  $\bar{G}_v(0)$  is monotonically decreasing and linear in  $v$ , and  $\bar{G}_{v_f(0)}(0) = 0$ .

Assume that (109) does not hold. Then  $G^*(\cdot)$  must cross  $\bar{G}_{v^*}(\cdot)$  from below at some  $\hat{w} \geq 0$ . But then, by following the fluid control for  $w \geq \hat{w}$  we get a feasible control with average cost  $v^*$ . Therefore,  $G^*(w) = \bar{G}_{v^*}(w)$  for  $w \geq \hat{w}$ , and hence  $G^*(w) \leq \bar{G}_{v^*}(w)$  for  $w \geq 0$  contradicting our assumption.

**Step 2:** Let  $G_W^*(w)$  denote the value function gradient for the optimal finite buffer control with workload buffer  $W$  and average cost  $v_W^*$ . Then

$$G_W^*(w) \leq G^*(w) \quad \text{for } w \in [0, W]$$

and hence  $G_W^*(w) \leq \alpha + \frac{w}{m\hat{\theta}}$ .

*Proof:* Compare the HJB equation for  $G^*(w)$  and  $G_W^*(w)$ :

$$\begin{aligned} v^* &= \min_{k \in [0, w/m_e]} (\Delta_k(w) - \theta(k)G^*(w)) + \frac{\sigma^2}{2}(G^*)'(w) \\ v_W^* &= \min_{k \in [0, w/m_e]} (\Delta_k(w) - \theta(k)G_W^*(w)) + \frac{\sigma^2}{2}(G_W^*)'(w) \end{aligned}$$

Now, for any  $w \geq 0$ , if  $G^*(w) = G_W^*(w)$ , then  $(G^*)'(w) \geq (G_W^*)'(w)$  since the terms in the parentheses are equal and  $v^* \geq v_W^*$ . That is,  $G_W^*(\cdot)$  can never cross  $G^*(\cdot)$  from below. Since  $G^*(0) = G_W^*(0)$ ,  $G^*(w) \geq G_W^*(w)$  for all  $w \geq 0$ .

**Step 3:** Define

$$T_W(w) \doteq \int_0^w \theta(k_W^*(u))du,$$

where  $k_W^*(w)$  is the optimal finite buffer control with workload buffer  $W$ . Then as  $W \rightarrow \infty$ ,  $T_W(W) = \Theta(W)$ . That is, the integral of the drift over the interval  $[0, W]$  for the family of controls  $k_W^*(\cdot)$  parameterized by  $W$  must asymptotically grow linearly in the buffer limit.

*Proof:* We begin by rewriting the HJB equation for  $G_W^*(w)$

$$\begin{aligned} \frac{\sigma^2}{2}(G_W^*)'(w) &= v_W^* - \min_{k \in [0, w/m_e]} (\Delta_k(w) - \theta(k)G_W^*(w)) \\ &\leq v_W^* - \frac{w}{m \vee m_e} + \theta(k_W^*(w))G_W^*(w). \end{aligned}$$

Take the integration

$$\int_{w=0}^W \frac{\sigma^2}{2}(G_W^*)'(w)dw \leq W \cdot v_W^* - \frac{W^2}{2(m \vee m_e)} + \left(\alpha + \frac{W}{m\hat{\theta}}\right) \int_0^W \theta(k_W^*(w))dw.$$

This implies

$$\frac{\sigma^2}{2}(G_W^*(W) - G_W^*(0)) \leq W \cdot v_W^* - \frac{W^2}{2(m \vee m_e)} + \left(\alpha + \frac{W}{m\hat{\theta}}\right) T_W(W).$$

The left-hand side of the above inequality is 0 (since the workload reflects at  $w = 0$  and  $w = W$ ). The first term on the right-hand side grows linearly in  $W$  since  $v_W^*$  is bounded by  $v^*$ . The second term grows as  $\Theta(W^2)$ . If  $T_W(W) = o(W)$  then the right-hand side becomes negative for  $W$  large enough – a contradiction. Therefore,  $T_W(W)$  must grow at least linearly. By our assumptions,  $\theta(w) \leq \hat{\theta} < \infty$ . Therefore,  $T_W(W) = \Theta(W)$ .

**Step 4:** Denote the density function of the workload under control  $k_W^*$  by  $f_W$ . The preceding step implies

$$f_W(W) = O(e^{-\beta W})$$

for some positive constant  $\beta > 0$ . That is, the density function for the workload under optimal finite buffer controls falls exponentially.

*Proof:* The density function is given by

$$\begin{aligned} f_W(w) &= \kappa_W e^{-\int_0^w \frac{\theta(k_W^*(u))}{\sigma^2/2} du} \\ &= \kappa_W e^{-\frac{T_W(w)}{\sigma^2/2}}, \end{aligned}$$

where  $\kappa_W$  is the normalization constant. Since  $T_W(W) = \Theta(W)$  by the preceding step,  $f_W(W) = O(e^{-\beta W})$  for some  $\beta > 0$ .

**Step 5:** Finally consider the following infinite buffer control, parameterized by workload  $W$ :

$$\tilde{k}_W(w) = \begin{cases} k_W^*(w) & w \leq W, \\ \hat{k} & w > W. \end{cases}$$

That is, we create a fluid continuation control with prefix  $k_W^*(w)$ . This results in a suboptimal infinite buffer control with average cost  $\tilde{v}_W \geq v^* \geq v_W^*$ . However, since the workload density decays exponentially under control  $k_W^*(\cdot)$ , and  $\theta(\hat{k}) > 0$ , the cost of  $\tilde{k}_W(\cdot)$  is at most  $O(e^{-\beta W})$  higher than  $v_W^*$ . That is

$$(v^* - v_W^*) \leq (\tilde{v}_W - v_W^*) = O(e^{-\beta W}).$$

■

**Proof of Proposition 6:** Recall the Newton-Raphson algorithm from Algorithm 2: We first pick a large enough value of workload  $W \geq \hat{k}m_e$  (which is not changed during subsequent iterations).

The goal of the Newton-Raphson algorithm then is to find the average cost of the optimal dynamic policy under the restriction that the control for  $w \geq W$  is the fluid control  $\hat{k}$ . With  $v_n$  as our guess in the  $n$ th iteration, we backwards evolve the ODEs:

$$v_n = \min_{k \in [0, w/m_e]} \left[ \frac{w}{m} + k \left( 1 - \frac{m_e}{m} \right) - \theta(k) G_{v_n}(w) \right] + \frac{\sigma^2}{2} G'_{v_n}(w) \quad (111)$$

$$1 = -\theta(k_{v_n}(w)) g_{v_n}(w) + \frac{\sigma^2}{2} g'_{v_n}(w) \quad (112)$$

for  $w \in [0, W]$  with (terminal) boundary conditions:

$$G_{v_n}(W) = \left( \hat{k} \left( 1 - \frac{m_e}{m} \right) - v_n + \frac{\sigma^2}{2\hat{\theta}} \right) \frac{1}{\hat{\theta}} + \frac{1}{m\hat{\theta}} W \quad (113)$$

$$g_{v_n}(W) = -\frac{1}{\hat{\theta}} \quad (114)$$

Here  $k_{v_n}(w)$  denotes the policy obtained while solving the ODE for  $G_{v_n}$ .

The updated guess for the  $(n+1)$ st iteration is

$$v_{n+1} = v_n - \frac{G_{v_n}(0)}{g_{v_n}(0)}.$$

We develop our proof of the proposition in several steps.

**Step 1 :**  $v_n \geq v_f(W)$  for  $n \geq 1$

*Proof:* Let  $\tilde{G}_v(w)$  (parameterized by  $v, w$ ) be given by the ODE

$$v = \frac{w}{m} + k_{v_n}(w) \left( 1 - \frac{m_e}{m} \right) - \theta(k_{v_n}(w)) \tilde{G}_v(w) + \frac{\sigma^2}{2} \tilde{G}'_v(w)$$

for  $w \in [0, W]$  with boundary condition

$$\tilde{G}_v(W) = \left( \hat{k} \left( 1 - \frac{m_e}{m} \right) - v + \frac{\sigma^2}{2\hat{\theta}} \right) \frac{1}{\hat{\theta}} + \frac{1}{m\hat{\theta}} W$$

This is essentially the same ODE as (111) but with the min operator replaced by the fixed policy  $k_{v_n}$ . The first observation is that  $\tilde{G}_v(0)$  is a linear function of  $v$ , and  $\tilde{G}_{v_n}(w) = G_{v_n}(w)$  for all  $w \in [0, W]$ . Further, denoting

$$\tilde{g}_v(w) = \frac{d}{dv} \tilde{G}_v(w)$$

it is easy to see that  $\tilde{g}_v(w) = g_{v_n}(w)$ . Therefore,

$$v_{n+1} = v_n - \frac{G_{v_n}(0)}{g_{v_n}(0)} = v_n - \frac{\tilde{G}_{v_n}(0)}{\tilde{g}_{v_n}(0)}$$

Since  $\tilde{G}_v(w)$  is a linear function in  $v$  for all  $w$ , the Newton-Raphson update for  $\tilde{G}_v(0)$  directly yields that value of  $v$  for which  $\tilde{G}_v(0) = 0$ . But this must be the average cost of policy  $k_{v_n}$ . Therefore,  $v_{n+1}$  is in fact the average cost of policy  $k_{v_n}$ . Since  $k_{v_n}$  is a feasible policy in the set  $\mathcal{F}_W$ , its average cost must be no less than  $v_f(W)$  and hence all the iterates  $\{v_1, v_2, \dots\}$  produced are larger than  $v_f(W)$ .

**Step 2:** The iterates for average cost  $\{v_1, v_2, \dots\}$  form a strictly decreasing sequence.

*Proof:* For this, we will show that  $G_v(0)$  is monotonically decreasing and Lipschitz continuous in  $v$  with derivative bounded away from 0. This would imply that for  $v > v_f(W)$ ,  $G_v(0) < 0$ , as well as  $g_v(0) < 0$ , and hence  $v_1 > v_2 > \dots > v_f(W)$ .

Consider  $v_a > v_b$ , and let  $G_a, g_a, k_a$  and  $G_b, g_b, k_b$  represent the solution of (111)-(114) and the optimal controls for  $v_a$  and  $v_b$ , respectively. Our goal is to show  $G_a(w) < G_b(w)$  for all  $w \geq 0$ . We rely on the following two facts:

1. Terminal condition:

$$G_b(w) - G_a(w) = \frac{a - b}{\hat{\theta}} \quad w \geq W$$

2. Bounds on  $G'_b(w) - G'_a(w)$  for  $w \in [0, W]$ :<sup>1</sup>

$$G'_b(w) - G'_a(w) = -\frac{2}{\sigma^2} \left[ (v_a - v_b) - \min_{k \in [0, w/m_e]} (\Delta_k(w) - \theta(k)G_a(w)) - \min_{k \in [0, w/m_e]} (\Delta_k(w) - \theta(k)G_b(w)) \right]$$

where recall that  $\Delta_k(w) = \frac{w}{m} + k \left(1 - \frac{m_e}{m}\right)$ . By the assumption  $G_a(w) \leq G_b(w)$  and Lemma 7,

$$-\frac{2}{\sigma} [(v_a - v_b) + d_\theta(G_b(w) - G_a(w))] \leq G'_b(w) - G'_a(w) \leq -\frac{2}{\sigma^2} [(v_a - v_b) - d_\theta(G_b(w) - G_a(w))].$$

Combining these two facts, we get

$$(v_a - v_b) \left[ \frac{1}{\hat{\theta}} + \frac{1}{d_\theta} \left( 1 - e^{-\frac{2d_\theta W}{\sigma^2}} \right) \right] \leq G_b(0) - G_a(0) \leq (v_a - v_b) \left[ \frac{1}{\hat{\theta}} + \frac{1}{d_\theta} \left( e^{\frac{2d_\theta W}{\sigma^2}} - 1 \right) \right]. \quad (115)$$

Thus we have proved that  $G_v(0)$  is monotonically decreasing in  $v$  and therefore the Newton-Raphson iterates will form a monotonically decreasing sequence. Further,  $G_v(w)$  is Lipschitz continuous in  $v$  for all  $w$ , and therefore its derivative with respect to  $v$  exists almost everywhere, and according to the first inequality of (115) this derivative is bounded away from 0.

The properties proved so far are sufficient to prove that the Newton-Raphson algorithm converges to the optimal  $v_f(W)$ , and the convergence rate is at least linear.

**Step 3:** The second derivative of  $G_v(0)$  with respect to  $v$  is finite in some neighborhood of  $v_f(W)$ , and hence the Newton-Raphson iterates converge quadratically to  $v_f(W)$ .

*Proof:* A sufficient condition to show a quadratic convergence rate for the Newton-Raphson method is that the first derivative is non-zero and the second derivative is finite in some neighborhood of the root. We have already shown that the first condition holds. That is  $g_v(0) > 0$  for all  $v$ . What remains to be done is to show that  $\frac{\partial^2 G_v(0)}{\partial v^2} = \frac{\partial g_v(0)}{\partial v}$  is finite in a neighborhood of  $v_f(W)$ . First consider the case where  $|1 - \frac{m_e}{m}| = 0$ , or in other words  $m_e = m$ . As pointed out before, the optimal policy in this case is the fluid policy for which the optimal average cost is found in the initialization phase itself and the algorithm terminates after one step. Therefore we assume  $|1 - \frac{m_e}{m}| > 0$  in the remainder of the proof.

Abbreviating  $\theta_a(w) \doteq \theta(k_a(w))$  and  $\theta_b(w) \doteq \theta(k_b(w))$ , consider the ODE for  $g_a(w)$ :

$$g'_a(w) = \frac{2}{\sigma^2} (1 + \theta_a(w)g_a(w))$$

---

<sup>1</sup>We use primes to denote derivatives with respect to  $w$ . Derivatives with respect to  $v$  are denoted with  $\frac{\partial}{\partial v}$  notation.

with boundary condition  $g_a(W) = -\frac{1}{\theta}$ . Therefore, we have

$$|g'_a(w) - g'_b(w)| = \frac{2}{\sigma^2} |\theta_a(w)g_a(w) - \theta_b(w)g_b(w)| \quad (116)$$

$$\leq \frac{2}{\sigma^2} (|\theta_a(w) - \theta_b(w)| (|g_a(w)| + |g_b(w)|) + 2d_\theta |g_a(w) - g_b(w)|). \quad (117)$$

As we have shown in the previous step,  $|G_b(w) - G_a(w)| \leq \kappa(w)|v_a - v_b|$  for some bounded function  $\kappa(w)$ . Combined with our regularity assumptions on  $\theta()$ , this implies

$$|\theta_a(w) - \theta_b(w)| \leq |G_b(w) - G_a(w)| \frac{S_\theta^3}{D_\theta \left|1 - \frac{m_e}{m}\right|} \leq |v_a - v_b| \frac{\kappa(w) S_\theta^3}{D_\theta \left|1 - \frac{m_e}{m}\right|}. \quad (118)$$

A rough justification of the first inequality is the following: Since

$$k_a(w) = \arg \min_{k \in [0, w/m_e]} \left( \frac{w}{m_e} - k \left(1 - \frac{m_e}{m}\right) + \theta(k)G_a(w) \right),$$

one of three cases occurs: (1)  $k_a(w) = 0$ , (2)  $k_a(w) = w/m_e$ , or (3)  $(1 - \frac{m_e}{m})/G_a(w) \in \partial\theta(k_a)$  where  $\partial\theta(k_a)$  is the set of subderivatives of  $\theta(k)$  at  $k = k_a$ . Consider the scenario where the third case occurs for  $k_a(w)$  as well as  $k_b(w)$  (other cases are easier to handle). By (59), the absolute value of the first derivative of  $\theta(k)$  is bounded by  $S_\theta$ . Therefore for the case  $0 < k_a(w) < \frac{w}{m_e}$  to occur  $|G_a(w)|, |G_b(w)|$  must be bounded away from 0. More precisely,  $\min(|G_a(w)|, |G_b(w)|) \geq \frac{|1 - \frac{m_e}{m}|}{S_\theta}$ . Now, by (60)  $D_\theta > 0$ , we must have

$$\left|1 - \frac{m_e}{m}\right| \left| \frac{1}{G_a(w)} - \frac{1}{G_b(w)} \right| \geq D_\theta |k_a(w) - k_b(w)|,$$

or

$$\begin{aligned} |k_a(w) - k_b(w)| &\leq \frac{|1 - \frac{m_e}{m}|}{D_\theta} \left| \frac{1}{G_a(w)} - \frac{1}{G_b(w)} \right| \\ &= \frac{|1 - \frac{m_e}{m}|}{D_\theta} \left| \frac{G_a(w) - G_b(w)}{G_a(w)G_b(w)} \right| \\ &\leq \frac{S_\theta^2}{D_\theta \left|1 - \frac{m_e}{m}\right|} |G_a(w) - G_b(w)|. \end{aligned}$$

Finally,

$$\begin{aligned} |\theta_a(w) - \theta_b(w)| &= |\theta(k_a(w)) - \theta(k_b(w))| \\ &\leq S_\theta |k_a(w) - k_b(w)| \\ &\leq \frac{S_\theta^3}{D_\theta \left|1 - \frac{m_e}{m}\right|} |G_a(w) - G_b(w)|. \end{aligned}$$

Finally substituting (118) into (117) gives

$$|g'_a(w) - g'_b(w)| \leq \frac{2}{\sigma^2} \left( |v_a - v_b| \frac{\kappa(w) S_\theta^3}{D_\theta \left|1 - \frac{m_e}{m}\right|} (|g_a(w)| + |g_b(w)|) + 2d_\theta |g_a(w) - g_b(w)| \right). \quad (119)$$

Now, a similar calculation to that in step 2 shows that  $g_v(w)$  is Lipschitz continuous in  $v$  for all  $w$ , and hence for  $w = 0$ . We believe the lower bound on  $\left| \frac{d^2\theta(k)}{dk^2} \right|$  we used to prove quadratic convergence is an artifact of our rather crude proof technique, and that quadratic convergence holds even without this restriction.  $\blacksquare$